Elisabeth Métais   Farid Meziane
Mohamad Saraee   Vijayan Sugumaran
Sunil Vadera (Eds.)

Natural Language Processing and Information Systems

**18th International Conference on Applications
of Natural Language to Information Systems, NLDB 2013
Salford, UK, June 2013, Proceedings**

 Springer

Lecture Notes in Computer Science 7934

Elisabeth Métais   Farid Meziane
Mohamad Saraee   Vijayan Sugumaran
Sunil Vadera (Eds.)

Natural Language Processing and Information Systems

18th International Conference on Applications
of Natural Language to Information Systems, NLDB 2013
Salford, UK, June 19-21, 2013
Proceedings

Springer

Volume Editors

Elisabeth Métais
Conservatoire National des Arts et Métiers
Paris, France
E-mail: metais@cnam.fr

Farid Meziane
Mohamad Saraee
Sunil Vadera
University of Salford
Salford, Lancashire, UK
Email: {f.meziane, m.saraee, s.vadera}@salford.ac.uk

Vijayan Sugumaran
Oakland University
Rochester, MI, USA
E-mail: sugumara@oakland.edu

Preface

This volume of *Lecture Notes in Computer Science* (LNCS) contains the papers presented at the 18th International Conference on Application of Natural Language to Information Systems, held at MediacityUK, University of Salford during June 19–21, 2013 (NLDB2013). Since its foundation in 1995, the NLDB conference has attracted state-of-the-art presentations and followed closely the developments of the application of natural language to databases and information systems in the wider meaning of the term.

The current conference proceedings reflect the development in the field and encompass areas such as sentiment analysis and mining, forensic computing, the Semantic Web and information search. This is in addition to the more traditional topics such as requirements engineering, question answering systems, and named entity recognition. NLDB is now an established conference and attracts researchers and practitioners from all over the world. Indeed, this year's conference saw submission for works using a large number of natural languages that include Chinese, Japanese, Arabic, Hebrew, and Farsi.

We received 80 papers and each paper was reviewed by at least three reviewers with the majority having four or five reviews. The Conference Co-chairs and Program Committee Co-chairs had a final consultation meeting to look at all the reviews and made the final decisions on the papers to be accepted. We accepted 21 papers as long/regular papers, 15 short papers, and 17 poster presentations.

We would like to thank all the reviewers for their time, effort, and for completing their assignments on time despite tight deadlines. Many thanks to the authors for their contributions.

June 2013

Elisabeth Métais
Farid Meziane
Mohamed Saraee
Vijay Sugumaran
Sunil Vadera

Organization

Conference Chairs

Elisabeth Métais   Conservatoire National des Arts et Metiers,
                          Paris, France
Farid Meziane   University of Salford, UK
Sunil Vadera   University of Salford, UK

Programme Committee Chairs

Mohamed Saraee   University of Salford, UK
Vijay Sugumaran   Oakland University Rochester, USA

Programme Committee

Jacky Akoka   CNAM, France
Frederic Andres   National Institute of Informatics, Japan
Apostolos Antonacopoulos University of Salford, UK
Eric Atwell   University of Leeds, UK
Abdelmajid Ben Hamadou   Sfax University, Tunisia
Bettina Berendt   Leuven University, Belgium
Johan Bos   Groningen University, The Netherlands
Goss Bouma   Groningen University, The Netherlands
Philipp Cimiano   Universität Bielefeld, Germany
Isabelle Comyn-Wattiau   ESSEC, France
Walter Daelemans   University of Antwerp, Belgium
Zhou Erqiang   University of Electronic Science and
                          Technology, China
Stefan Evert   University of Osnabrück, Germany
Vladimir Fomichov   National Research University Higher School
                          of Economics Russia
Alexander Gelbukh   Mexican Academy of Science, Mexico
Jon Atle Gulla   NTNU, Norway
Karin Harbusch   Koblenz University, Germany
Dirk Heylen   University of Twente, The Netherlands
Helmut Horacek   Saarland University, Germany

Ashwin Ittoo   Groningen University, The Netherlands
Paul Johannesson   Stockholm University, Sweden
Sophia Katrenko   Utrecht University, The Netherlands
Epaminondas Kapetanios   University of Westminster, UK
John Kean   University of Manchester, UK
Zoubida Kedad   Université de Versailles, France
Christian Kop   University of Klagenfurt, Austria
Valia Kordoni   Saarland University, Germany
Leila Kosseim   Concordia University, Canada
Zornitsa Kozareva   University of Southern California, USA
Nadira Lammari   CNAM, France
Dominique Laurent   University of Cergy Pontoise, France
Jochen Leidner   Thomson Reuters, USA
Piroska Lendvai   Hungarian Academy of Sciences, Hungary
Johannes Leveling   Dublin City University, Ireland
Deryle Lonsdale   Brigham Young Uinversity, USA
Rob Malouf   San Diego State University, USA
Farhi Marir   London Metropolitan University, UK
Heinrich C. Mayr   University of Klagenfurt, Austria
Farid Meziane   Salford University, UK
Elisabeth Métais   CNAM, France
Marie-Jean Meurs   Concordia University, Canada
Luisa Mich   University of Trento, Italy
Shamima Mithun   Concordia University, Canada
Marie-Francine Moens   Katholieke Universiteit Leuven, Belgium
Andres Montoyo   Universidad de Alicante, Spain
Rafael Muñoz   Universidad de Alicante, Spain
Guenter Neumann   DFKI, Germany
Jan Odijk   Utrecht University, The Netherlands
Jim O'Shea   Manchester Metropolitan University, UK
Karim Ouazzane   London Metropolitan University, UK
Pit Pichappan   Al Imam University, Saudi Arabia
Shaolin Qu   Michigan State University, USA
Mike Rosner   University of Malta, Malta
German Rigau   University of the Basque Country, Spain
Fabio Rinaldi   University of Zurich, Switzerland
Mathieu Roch   Université Montpellier 2, France
Patrick Saint-Dizier   Université Paul Sabatier, France
Mohamed Saraee   University of Salford, UK
Hide Sasaki   Ritsumeikan University, Kyoto, Japan
Khaled Shaalan   The British University in Dubai, UAE
Max Silberztein   Université de Franche-Comté, France

Samira Si-Said Cherfi	CNAM, France
Veda Storey	Georgia State University, USA
Vijayan Sugumaran	Oakland University Rochester, USA
Bernhard Thalheim	Kiel University, Germany
Michael Thelwall	University of Wolverhampton, UK
Krishnaprasad Thirunarayan	Wright State University, USA
Juan Carlos Trujillo	Universidad de Alicante, Spain
Christina Unger	Universität Bielefeld, Germany
Alfonso Ureña	University of Jaén, Spain
Sunil Vadera	University of Salford, UK
Panos Vassiliadis	University of Ioannina, Greece
Robert Wagner	Linz University, Austria
René Witte	Concordia University, Canada
Magdalena Wolska	Saarland University, Germany
Bing Wu	Dublin Institute of Technology, Republic of Ireland
Jim Yip	University of Salford, UK

Table of Contents

Full Papers

Short Papers

Poster Papers

Extraction of Statements in News for a Media Response Analysis

Thomas Scholz and Stefan Conrad

Heinrich-Heine-University, Institute of Computer Science, Düsseldorf, Germany
{scholz,conrad}@cs.uni-duesseldorf.de

Abstract. The extraction of statements is an essential step in a Media Response Analysis (MRA), because statements in news represent the most important information for a customer of a MRA and can be used as the underlying data for Opinion Mining in newspaper articles. We propose a machine learning approach to tackle this problem. For each sentence, our method extracts different features which indicate the importance of a sentence for a MRA. Classified sentences are filtered through a density-based clustering, before selected sentences are combined to statements. In our evaluation, this technique achieved better results than comparison methods from Text Summarization and Opinion Mining on two real world datasets.

Keywords: Information Extraction, Text Data Mining, Media Response Analysis, Text Summarization, Opinion Mining.

1 Motivation

Many organisations such as companies, political parties, institutes or foundations have to care for their public relation. So, they need to analyse the success in PR activities or the media's image about themselves or their products. To obtain this information, they have to perform a Media Response Analysis (MRA) [14]. But a MRA means a big human effort. Media analysts (professional experts in the field of a MRA) have to read approximately 200 to 800 news articles each week and extract relevant statements for the customers.

The extraction of relevant statements is essential for a MRA [10,14]. These statements contain the most important information of a news article for the customer of a MRA [14]. In addition, the statements can be used for Opinion Mining in newspaper articles [9,11]. For a MRA, it is much less important to analyse the tonality [10] of whole documents. We want to demonstrate that and the role of statements in the following: Figure 1 shows a translated example of the pressrelations dataset [10]. It represents a publicly available dataset[1] of a MRA about the two biggest political parties in Germany in order to train and evaluate methods for this task. The underlined passages are statements of the gold standard, extracted by two professional media analysts. They contain

---

[1] http://www.pressrelations.de/research/

<headline>Greenpeace: Platzeck blocks energy turnaround</headline>

<text>Potsdam (ots) – Today Greenpeace activists protest against the climate-damaging direction of the Brandenburg prime minister Matthias Platzeck (SPD) on closed meeting of the SPD leadership in front of the island hotel Hermannswerder in Potsdam. The activists pile up 20 tons of lignite on the approach road and hold a banner "Dear SPD, Platzeck's lignite blocks the energy turnaround". In its concept of energy strategy 2030, the red-red government of Brandenburg still put on lignite which is the most climate-damaging energy source.

"The actual concept of energy strategy leads in a impasse for climate policy", says Greenpeace's energy expert Anike Peters. "It is a mistake to think that there will be an European infrastructure for carbon dioxide and in the same way lignite will be burnt. Prime minister Platzeck harms his country, if he continues to ignore the advantages of the phase-out of lignite. Renewable energy could generate more jobs and bring a greater value added into the country", says Peters. [...]

Greenpeace demands Mr. Platzeck to extend period for a statement to six weeks, as soon as it is possible to view the complete content of any related studies.

Greenpeace study: Lignite phase out promises benefits for Brandenburg

A new Greenpeace study shows that the number of jobs can increase from 11,500 today to more than 19,000 employees in the year 2030. [...] </text>

Fig. 1. A translated example [10] of a news article with statements

the most important information for the SPD (the governing party of the region Brandenburg) and both are annotated with a negative sentiment. The marked sentences are not relevant for another party, e.g., and for Greenpeace other sentences are relevant: a relevant statement would be the last two sentences of the text snippet. So, the results of MRA depends on the analysis objects (in general the customer of a MRA and its competitors or in the case of the pressrelations dataset the German parties SPD and CDU). In this paper, we want to concentrate on the extraction of relevant statements, because they are essential for a MRA and also other approaches show that a well-considered selection of text parts could improve Sentiment Analysis for opinion-bearing text [8] or even work with statements [11].

Task Definition: *Let $d \in D$ be a document and D a collection of news articles. The task is to find a partition P of the set of all sentences S_d for every $d \in D$ so that P has ν elements and $\nu - 1$ elements are relevant statements (ν is unknown before analysis).*

$$f_p : d \mapsto P = \{p_1, ..., p_\nu\} = \underbrace{\{\{s_j, s_{j+1}, ...\}, ..., \{s_k, s_{k+1}, ...\}}_{\nu-1}, \{s_l, s_m, ...\}\} \quad (1)$$

p_ν contains all not relevant sentences and all p_i with $i \in \{1, ..., \nu - 1\}$ include the relevant statements. A statement is a consecutive sequence of relevant sentences (a statement usually consists of up to four sentences). In general, documents with only one element (all sentences are not relevant) and elements with only one sentence ($p = \{s_i\}$, e.g.) are possible.

As figure 1 shows, the relevant statements are not only sentences, in which the certain search strings (such as 'SPD', 'Platzeck', or 'Greenpeace') appear. Sometimes a coreference resolution is needed (cf. the last sentence in the first

statement), but sometimes even such resolution would not help (cf. the last sentence in the second statement). In our evaluation we will show that this is often the case. Moreover, the antepenultimate sentence contains the word 'Platzeck' and is not relevant, because it contains only additional information. So, we propose a machine learning technique which is based on significant features of relevant sentences and filter misclassified sentences by a density-based clustering.

The rest of the paper is organized as follows: We discuss Related Work in the next section. In section three we explain our machine learning-based method for the statement extraction. We evaluate and compare the results of our approach with other techniques in section 4, before we conclude in the last section.

2   Related Work

The extraction of relevant statements for a MRA is related to several kinds of areas: the automated creation of Text Summaries [1,6,7,12], Information Extraction [3,13] and Opinion Mining [8,9,11].

Automatic Text Summaries have a long history. An early approach works with coreference chains [1] to estimate the sentences of a summary. Turney extracts important phrases by learned rules [12], while Mihalcea and Tarau build graphs using Page Rank and a similarity function between two sentences [7]. A language-independent approach for Text Summarization proposed by Litvak et al. [6] is called **DegExt**. The approach transforms a given text into a graph representation where words become nodes. Within this graph, the important words are estimated by nodes with a high connectivity. These words are extracted as keywords of the text and the summary consists of all sentences which contain keywords. They report better results than TextRank [7] and GenEx [12] on the benchmark corpus of summarized news articles of the 2002 DUC by extracting 15 keywords. So, we took DegExt as one of our comparison methods.

Also, the task of this contribution is related to Information Extraction tasks such as the extraction of statements for market forecasts [13]. Here, a statement consists of a 5-tuple of topic, geographic scope, period of time, amount of money or growth rate, and the statement time, whereas the relation of time and money information is particularly important. Hong et al. [3] extract events from sentences. The event extraction covers the determination of the type of the event, its participants, and their role. Both definitions of statements/events and their methods do not fit in our issue.

In the field of Opinion Mining, the identification of Opinion Holders is an important task [4]. But in a MRA, we know the objects (organisations or persons, e.g.) of an analysis. But the automatic extraction of statements is very interesting for Opinion Mining tasks [10], to classify the tonality [11] as well as the viewpoints of extracted statements [9]. The approach [8] of Sarvabhotla et al., called **RSUMM** (**R**eview **Summ**ary), creates summaries of reviews for Opinion Mining tasks. They weight sentences by the importance of the containing words and the subjectivity. In this way, they select the most important and subjective sentences for their subjective excerpt [8]. We apply two variants of this approach for our evaluation.

In the news domain, the MPQA corpus [15] is a very important test corpus for issues of Opinion Mining. Unfortunately, it has no extracted statements, because it is not designed as a MRA. The pressrelations dataset [10] is a publicly available corpus of a MRA. It contains 617 news articles which contain 1,521 statements for the two biggest political parties in Germany. Overall, the articles include 15,089 sentences from which 3,283 are relevant for the two political parties. This dataset is part of our evaluation. To evaluate our approaches, we use metrics of the Text Summarization area, because this field has several things in common with our task. Lin [5] proposes widely acknowledged metrics to estimate the quality of text summaries. We use the ROUGE-L score to determine the quality of the extracted statements.

3   A Machine Learning-Based Method for Statement Extraction

3.1   Learning Relevant Sentences

As shown in the examples (figures 1 and 2), statements are not just consecutive sentences or whole paragraphs, which contain certain search strings such as the name of a person or a party. In figure 2, the last sentences of each statement do not contain a keyword such as 'SPD'.

<headline>Work of the Internet-Enquete is not yet finished</headline>

<text>Lars Klingbeil, who is the net-political speaker of the SPD Parliamentary Group, responds to the declaration of the Union fraction about the Enquete-Commission "Internet and digital society":

The establishment of a committee about network policy and digital society is right. The SPD Parliamentary Group understands network policy as fundamental and comprehensive approach which has to be reflected in different fields of politics. Network policy is social policy.

Therefore, the aim must be, that net policy is anchored in every committees of the Bundestag prominently.

The establishment of net political issues needs time, so for a transitional period we need a independent and full rights main-committee about net policy and digital society, which in charge of these topics in the Bundestag. Therefore, the SPD Parliamentary Group pressed for reinstatement of the subcommittee "new media" against the resistance of the Union at the beginning of the legislative period. Even then we demanded it as a proper committee.

It is correct that this new main-committee will be established after the end of the Internet-Enquete. [...] </text>

Fig. 2. Second translated example of an annotated news item [10]

We propose an approach based on machine learning for the extraction of relevant statements. Thereby, we consider input (new texts) as a sequence of sentences, and we decide for every sentence: Is this sentence relevant or not. For this task, we extract different features (cf. table 1) which indicate the importance of a sentence for a MRA: First, we count the number of important persons and organisations in sentences (an organisation could also be a product; some funds

are a product and a (subsidiary) company, e.g.). In a MRA, some people are of particular importance, because they are press spokesmen/spokeswomen of a relevant organisation (the customer's company or competitor) or they are an advertising medium. Media analysts collect list of these entities in so-called codebooks [9], because it is very difficult for humans to remember all relevant persons and organisations.

For a Named Entity Recognition (NER), we apply GATE (General Architecture for Text Engineering)[2] to extract the persons and organisations in the text. We have designed new JAPE Rules[3] to improve our NER. The new rules handle all important entities from our codebook with the highest priority. This secures that these entities are found with a very high probability. Furthermore, we improved the coreference resolution by adding a German pronominal coreferencer. We divided our list into three parts: female person, male person and neuter. In this way, we got the gender information for our NER. For Part-Of-Speech Tagging and lemmatisation, we use the TreeTagger[4].

Table 1. Feature set for our SVM classifier and our density-based clustering

Classification Features:	Clustering Features:
Entity Features:	*Word Features:*
k_1: number of important organisations	$(c_1, c_2, ..., c_{|V|})$:
k_2: number of important persons	the frequencies of words in
k_3: number of organisations	the statements classified
k_4: number of persons	as relevant
Importance of Words Features:	$(c_i =$ frequency of term i
k_5: number of headline words	in the sentence; V is the set of
k_6: average tf-idf score	all terms in all relevant sentences)

We count all elements of the coreference chains which belong to (important) persons and organisations/products. So, we call k_1 the number of important organisations and k_2 the number of important persons, while k_3 is the number of all organisations and k_4 the number of all persons in one sentence (cf. table 1). Also, it is significant how many headline words appear in the sentence. Headlines of news articles contain often compressed information about the whole article and so an occurrence of headline words can indicate a relevant statement. Therefore, k_5 is the number of headline words in a statement. Likewise, the statements themselves reflect important information about the article. For this reason, we measure the average tf-idf score of all words in the sentence (k_6 in table 1).

For the classification, we use a SVM[5]. We force the SVM to balance the performance on the two classes through the *balance cost* parameter, because there are many more irrelevant instances (cf. section 4.2).

---

[2] GATE: http://gate.ac.uk/

[3] Developing Language Processing Components with GATE Version 6 (a User Guide): http://gate.ac.uk/

[4] TreeTagger: http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[5] Rapidminer standard implementation (http://rapid-i.com/).

3.2   Filtering by Density-Based Clustering

After the classification of each sentence, we select all sentences which are classi-
fied as relevant. For every sentence, we count the frequency of every word in the
sentence and use these frequencies as input features (cf. table 1) for the perfor-
mance of a DBSCAN clustering [2]. We set the parameter *Eps* [2] (the radius of
a neighbourhood) to 1.0 and *MinPts* [2] (the number of minimum points in an
Eps neighbourhood) is set to 2. This secures that the clusters are very similar
and at the same time a similar misclassification occurs at least three times.

In a clustering approach, the clusters are usually more interesting in order
to identify objects, which share commonalities. But statements are representing
many different information and opinions over a large document corpus [14]. So,
our approach works the other way round and filters out clusters of not relevant
sentences because really relevant sentences tend to be noise and the same clas-
sification mistakes appear several times and thus become clusters. Thereby, we
use only sentences which are noise from a clustering perspective (cf. next sec-
tion). Since only the sentences classified as relevant are used for our clustering,
computational time can be saved for the performance of the clustering.

3.3   Statement Extraction Step

Our technique combines sentences which are classified as relevant by our SVM
and do not belong to any cluster in DBSCAN clustering. The input parameters
of the algorithm are the set of all sentences, the calculated classification model,
and the calculated clustering model:

Combine Sentences to Statements

```
generateStatements(Sentences S, Classification Model K,
      Clustering Model C){
  create empty List of Statements R;
  for each s in S do
    if K(s) = RELEVANT and C(s) = NOISE
      then add s in R;
  for each r1 in R
    for each r2 in R
      if(r1.EndOffset + 1 = r2.StartOffset)
        remove r1 and r2 from R;
        add (combine(r1,r2)) in R;
  return R;
}
```

The method `combine` takes two consecutive statements and append the second
one to the first one. `R` contains all p_i with $i \in \{1, ..., \nu-1\}$ and p_ν are all sentences
which are not a part of an element in `R`.

4   Evaluation

4.1   Experimental Setup

The Text Summarization method DegExt [6] is very language-independent, because the only required NLP resource is a tokenizer. DegExt allows to choose the number of keywords (referred to as N) and, as a consequence, the size of the summaries. We test several values for N, because the results of the experiments of Litvak et al. show that the choice of N is important for the quality of the result [6]. Consecutive sentences of a summary are combined into a statement.

We evaluate the RSUMM method [8] in two variants: The 'classical' method (denoted as RSUMM X%) calculates the lexical similarity between each sentence as a vector and the vectors of the most important words or the most subjective terms, respectively [8]. They compute a final score by adding the Jaccard similarity of both scores [8] and select the top X % of the sentences which got the highest scores. We use 20% of our training examples to create the vectors *adf* (average document frequency) and *asm* (average subjective measure) [8].

As a second variant, we use both RSUMM scores as input values for a classifier (denoted as RSUMM(+SVM)) and classify every sentence. Sarvabhotla et al. use the SVMLight package[6], so we apply this learner. But we obtain a very low accuracy (16.43% by using 50% for training, e.g.), because the classifier tends to qualify every sentence as relevant. As a consequence, we use the SVM of our technique which achieved better results (cf. section 4.2).

As two other baselines, we construct simple bags of words for every sentence to classify the sentences by our classifier (denoted as TSF-Matrix 5%, where TSF stands for term sentence frequency and the size of the training data is 5%). Likewise, we use only the extracted coreference chains of our important entities to identify statements (denoted as Coreference Chains): If one element of a chain of an important entity appears in the sentence, the sentence is relevant and consecutively relevant sentences are combined to statements.

We test the methods on two datasets: The pressrelations dataset [10] has 617 articles with 1,521 gold statements and an own dataset with 5,000 articles of a MRA about a financial service provider and 4 competitors (called **Finance**). The articles include 7,498 statements. The codebook for the finance dataset includes 384 persons, 19 organisations, and 10 products, while the codebook for the pressrelations dataset contains 386 persons (all party members of the 17th German Bundestag[7], the German parliament), and 18 entries of organisations (names and synonyms of the parties and concepts such as 'government' or 'opposition' [9]). The same codebooks are used in [9].

4.2   Results

For the step of learning relevant sentences, table 2 and 3 show the results for classifying single sentences as relevant or not. As the tables show, our classifier

---

[6] http://svmlight.joachims.org/
[7] collected from http://www.bundestag.de

needs only very limited training data (5% or 0.5%, resp.) to obtain good results (there is nearly no difference between using 15% or 5% on the pressrelations dataset). On Finance, the classifier requires even less data for good results. The results show that it is more difficult to identify the relevant sentences, while precision and recall of not relevant examples are very high. One reason is unequal distribution of the two classes, of course. Finance includes 13,084 relevant sentences and 145,219 not relevant sentences. However, the tables show that our method achieves better results on sentence level than RSUMM (+SVM).

Table 2. Results of the sentence classification on the pressrelations dataset

Method	Data for Training	Accuracy	Not Relevant		Relevant	
			Precision	Recall	Precision	Recall
RSUMM (+SVM)	2.5%	0.6403	0.2591	0.5923	0.8854	0.6503
RSUMM (+SVM)	5%	0.6938	0.8579	0.7556	0.2501	0.3943
RSUMM (+SVM)	10%	0.659	0.8659	0.6969	0.2434	0.4246
RSUMM (+SVM)	15%	0.6525	0.8661	0.6866	0.2443	0.4882
our approach	2.5%	0.4918	0.8315	0.4872	0.1694	0.5143
our approach	5%	0.8172	0.8912	0.8885	0.4597	0.4667
our approach	10%	0.8178	0.8914	0.8892	0.4601	0.4656
our approach	15%	0.8173	0.8111	0.8890	0.4479	0.4633

Table 3. Results of the sentence classification on Finance

Method	Data for Training	Accuracy	Not Relevant		Relevant	
			Precision	Recall	Precision	Recall
RSUMM (+SVM)	0.25%	0.6883	0.941	0.7108	0.0831	0.3702
RSUMM (+SVM)	0.5%	0.7067	0.9407	0.7321	0.0843	0.3481
RSUMM (+SVM)	1%	0.7579	0.9395	0.7917	0.0872	0.2807
RSUMM (+SVM)	5%	0.7075	0.9408	0.7329	0.0847	0.3488
our approach	0.25%	0.5045	0.954	0.4931	0.0853	0.6653
our approach	0.5%	0.9296	0.9575	0.9675	0.4641	0.3958
our approach	1%	0.9072	0.9614	0.9383	0.3514	0.4698
our approach	5%	0.9073	0.9618	0.9384	0.3515	0.4704

For our further experiments, we use only 5% on the pressrelations dataset and 0.5% on the finance dataset for training, because these values achieve good results and, for a practical solution, a technique should require less training as possible. Here, we measure how many statements match the annotated statements of the both datasets (denoted as Gold Standard Match). As well, we use the ROUGE-L score [5] which is based on the idea that two summaries are similar, if the size of the longest common subsequence (LCS) [5] is large:

$$R_{lcs} = \frac{LCS(X,Y)}{m} \qquad P_{lcs} = \frac{LCS(X,Y)}{n} \qquad F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \quad (2)$$

X is the annotated statement of the dataset, Y is the candidate statement, m is the length (in characters) of the gold statement X and n is the length of Y. The typical ROUGE-L score is the LCS-based F-measure, where β is set to a very high number and therefore the F-measure only depends upon R_{lcs}. We proceed not in the same way, because we are also interested in a high precision (a wrong statement can falsify the results of a MRA or means more effort to check the results). Therefore, we set $\beta = 1$ for F_{lcs}, but we also report the R_{lcs} and P_{lcs} values, too.

The results of the final generation of statements are shown in table 4 and 5. The results of our approach are listed in two lines: The first line shows the results without the clustering step (denoted as our approach), which is added for the results of the second line (+ clustering). Our method achieves the best F-measure value for the identification of the perfect match of the gold statements and at the same time the best ROUGE-L values on both dataset. The F-score of the gold match is an improvement of over 7% or 14%, resp., in comparison with the second best method (RSUMM 10%). The F_{lcs} values are over 20% or 27% higher than our TSF-Matrix. The results show that the clustering can increase the results, especially on the pressrelations dataset by over 2%. The DegExt method is most effective by using N=6 or N=5 on the pressrelations or Finance dataset, respectively. DegExt obtains a F-measure of 8.55% or 2.57% of the gold standard and the score F_{lcs} is 29.69% or 19.31%, respectively. RSUMM achieved better F-scores than DegExt in the match of the gold standard, but the ROUGE-L scores of F_{lcs} are nearly the same. The parameter X has less effect on both F-scores (a higher X value increases recall in the same way it decreases precision). The results show that a coreference resolution (as a preprocessing step of our approach) achieves partially precise results, but it only finds a smaller proportion of relevant statements.

But how important is a perfect match of the gold statements? If we take a look at figure 1 and 2, it is even for humans hard to decide, where a statements starts or ends. In many cases (as in example 1 and 2) it is not important, if a statement starts one sentence earlier or ends one sentence later which is often the case for the extracted statements (the ROUGE-L scores show this, e.g.). This is the reason for the low percentage values of the recall, but what is the reason for low precision values? Are so many machine-generated statements not relevant? The most approaches tend to extract more statements as in the gold annotation and we perform a deeper analysis of extracted statements in the next section.

4.3   Profound Analysis of the Extracted Statements

In examining the reprocessing issue in detail, the high precision of the ROUGE-L score and the low precision in the match with the gold statements is remarkable. On the one hand, the method can find the most important information in statements, but on the other hand, why did this technique (and most of the other

Table 4. Results of the statement extraction on the pressrelations dataset

Method	Extracted Statements	Gold Standard Match			ROUGE-L		
		Prec	Rec	F1	P_{lcs}	R_{lcs}	F_{lcs}
DegExt (N=1)	758	0.0752	0.0375	0.05	0.384	0.1278	0.1918
DegExt (N=2)	1,349	0.0801	0.071	0.0753	0.3929	0.2092	0.273
DegExt (N=3)	1,869	0.0776	0.0953	0.0855	0.3749	0.2603	0.3073
DegExt (N=5)	2,679	0.0646	0.1137	0.0824	0.3382	0.3085	0.3227
DegExt (N=6)	2,948	0.0648	0.1256	0.0855	0.332	0.327	0.3295
DegExt (N=7)	3,141	0.0592	0.1223	0.0798	0.3241	0.3324	0.3282
DegExt (N=8)	3,246	0.0564	0.1203	0.0768	0.3196	0.3348	0.327
DegExt (N=10)	3,358	0.0497	0.1098	0.0685	0.3172	0.3301	0.3235
DegExt (N=15)	3,338	0.0419	0.092	0.0576	0.3081	0.2988	0.3034
RSUMM (5%)	725	0.1807	0.0889	0.1192	0.3345	0.1734	0.2284
RSUMM (10%)	1,359	0.1405	0.1297	0.1349	0.3152	0.2761	0.2944
RSUMM (15%)	1,971	0.1152	0.1541	0.1318	0.2893	0.3513	0.3173
RSUMM (20%)	2,587	0.0928	0.1629	0.1182	0.2665	0.4171	0.3252
RSUMM (25%)	3,243	0.082	0.1806	0.1128	0.2438	0.4874	0.325
RSUMM(+SVM)	3,200	0.0816	0.1716	0.1115	0.2452	0.4776	0.324
TSF-Matrix 5%	2,321	0.0866	0.1321	0.1046	0.363	0.3399	0.3511
Coreference Chains	891	0.1852	0.1085	0.1368	0.5551	0.2778	0.3703
our approach	2,233	0.1536	0.2258	0.1828	0.5545	**0.4976**	0.5245
+ clustering	1,841	**0.1896**	**0.2302**	**0.2079**	**0.6302**	0.4951	**0.5545**

Table 5. Results of the statement extraction on the Finance dataset

Method	Extracted Statements	Gold Standard Match			ROUGE-L		
		Prec	Rec	F1	P_{lcs}	R_{lcs}	F_{lcs}
DegExt (N=1)	5,630	0.0238	0.0179	0.0204	0.2205	0.1077	0.1447
DegExt (N=2)	10,720	0.0207	0.0296	0.0244	0.2062	0.1655	0.1836
DegExt (N=3)	15,159	0.0181	0.0367	0.0242	0.1955	0.2042	0.1998
DegExt (N=4)	18,752	0.0175	0.0439	0.025	0.1883	0.2332	0.2084
DegExt (N=5)	21,724	0.0173	0.0501	0.0257	0.1826	0.2528	0.212
DegExt (N=6)	24,022	0.0165	0.0528	0.0251	0.1769	0.2628	0.2115
DegExt (N=7)	25,899	0.016	0.0552	0.0248	0.1725	0.2673	0.2097
DegExt (N=10)	29,518	0.0143	0.0561	0.0228	0.1655	0.2739	0.2063
DegExt (N=15)	32,117	0.0136	0.0583	0.0221	0.1596	0.2707	0.2008
RSUMM (5%)	8,214	0.0435	0.0507	0.0468	0.1775	0.1749	0.1762
RSUMM (10%)	15,090	0.0366	0.0784	0.0499	0.1649	0.2472	0.1978
RSUMM (15%)	21,905	0.0305	0.095	0.0462	0.1531	0.3045	0.2038
RSUMM (20%)	28,588	0.0271	0.1101	0.0435	0.1449	0.3524	0.2054
RSUMM (25%)	35,498	0.0243	0.1226	0.0406	0.1373	0.4023	0.2047
RSUMM(+SVM)	54,339	0.004	0.0312	0.0071	0.11	0.2343	0.1497
TSF-Matrix 5%	37,105	0.0258	0.129	0.043	0.2068	0.3877	0.2697
Coreference Chains	5,378	**0.1991**	0.1428	0.1663	0.6059	0.3572	0.4494
our approach	7,937	0.1713	0.2176	0.1917	**0.6312**	0.4754	0.5423
+ clustering	7,899	0.1707	**0.2212**	**0.1927**	0.6295	**0.4846**	**0.5476**

methods) tend to extract more statements than the number of gold statements? Two media analysts examined all extracted statements on the pressrelations dataset in a blind study (they do not know the extraction method) and reconsider all extracted statements: A statement is correct, when it is relevant (for the analysis objects) and a tonality [10] with a viewpoint [9] can be estimated. We use all methods with the best parameters (based on F_{lcs}).

Table 6. Results of reconsidering the statement extraction on the pressrelations dataset

Method	Precision	Recall	F-score
DegExt (N=6)	0.4156	0.7076	0.5236
RSUMM (20%)	0.4846	0.7433	0.5867
our approach + clustering	0.7968	0.8499	0.8225

The findings are depicted in table 6. Here, the F-score is almost 24% higher than the second best approach (RSUMM (20%)). This analysis shows that the approach extracts many more relevant statements which are not part of the gold annotation. There are several reasons for this: In a MRA [14] sometimes only a number of top-N statements are used. So, besides the gold statements which are found exactly or partially, the machine-based approaches find more statements, which are less important, but nevertheless adequate statements. Furthermore, many of these statements are neutral, so that they are not all extracted, because too many neutral statements may dilute the tonality in a practical analysis.

5   Conclusion

Our approach outperforms all comparison methods on both datasets. The findings point out that the extraction of statements for a MRA could not be solved only by Text Summarization. Furthermore, our evaluation shows that our technique can find many adequate statements. On the one hand, this approach can be utilized to help media analysts who could save time by extracting relevant statements. And on the other hand, our method closes a gap in an automated approach for a MRA, because the combination of this approach, the classification of the tonality [10,11] and the determination of perspectives [9] represents a fully automated generation of analysis data for a MRA.

References

1. Azzam, S., Humphreys, K., Gaizauskas, R.: Using coreference chains for text summarization. In: Proc. of the Workshop on Coreference and its Applications, CorefApp 1999, pp. 77–84 (1999)

2. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of the 2nd Intl. Conf. on Knowledge Discovery and Data Mining (KDD 1996), pp. 226–231 (1996)
3. Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., Zhu, Q.: Using cross-entity inference to improve event extraction. In: Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT 2011, vol. 1, pp. 1127–1136 (2011)
4. Kim, S.-M., Hovy, E.: Extracting opinions, opinion holders, and topics expressed in online news media text. In: Proc. of the Workshop on Sentiment and Subjectivity in Text, SST 2006, pp. 1–8 (2006)
5. Lin, C.-Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out: Proc. of the ACL 2004 Workshop, pp. 74–81. Association for Computational Linguistics (2004)
6. Litvak, M., Last, M., Aizenman, H., Gobits, I., Kandel, A.: Degext - a language-independent graph-based keyphrase extractor. In: Proc. of the 7th Atlantic Web Intelligence Conference (AWIC 2011), pp. 121–130 (2011)
7. Mihalcea, R., Tarau, P.: TextRank: Bringing order into texts. In: Proc. of the 2004 Conf. on Empirical Methods in Natural Language Processing, EMNLP 2004 (2004)
8. Sarvabhotla, K., Pingali, P., Varma, V.: Sentiment classification: a lexical similarity based approach for extracting subjectivity in documents. Inf. Retr. 14(3), 337–353 (2011)
9. Scholz, T., Conrad, S.: Integrating viewpoints into newspaper opinion mining for a media response analysis. In: Proc. of the 11th Conf. on Natural Language Processing, KONVENS 2012 (2012)
10. Scholz, T., Conrad, S., Hillekamps, L.: Opinion mining on a german corpus of a media response analysis. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2012. LNCS, vol. 7499, pp. 39–46. Springer, Heidelberg (2012)
11. Scholz, T., Conrad, S., Wolters, I.: Comparing different methods for opinion mining in newspaper articles. In: Bouma, G., Ittoo, A., Métais, E., Wortmann, H. (eds.) NLDB 2012. LNCS, vol. 7337, pp. 259–264. Springer, Heidelberg (2012)
12. Peter, D.: Turney. Learning algorithms for keyphrase extraction 2(4), 303–336 (2000)
13. Wachsmuth, H., Prettenhofer, P., Stein, B.: Efficient statement identification for automatic market forecasting. In: Proc. of the 23rd International Conference on Computational Linguistics, COLING 2010, pp. 1128–1136 (2010)
14. Watson, T., Noble, P.: Evaluating public relations: a best practice guide to public relations planning, research & evaluation. PR in practice series, ch. 6, pp. 107–138. Kogan Page (2007)
15. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language Resources and Evaluation 39(2-3), 165–210 (2005)

Sentiment-Based Ranking of Blog Posts Using Rhetorical Structure Theory

Jose M. Chenlo[1], Alexander Hogenboom[2], and David E. Losada[1]

[1] Centro de Investigación en Tecnoloxías da Información (CITIUS)
Universidad de Santiago de Compostela, Spain
{josemanuel.gonzalez,david.losada}@usc.es
[2] Econometric Institute
Erasmus University Rotterdam, The Netherlands
hogenboom@ese.eur.nl

Abstract. Polarity estimation in large-scale and multi-topic domains is a difficult issue. Most state-of-the-art solutions essentially rely on frequencies of sentiment-carrying words (e.g., taken from a lexicon) when analyzing the sentiment conveyed by natural language text. These approaches ignore the structural aspects of a document, which contain valuable information. Rhetorical Structure Theory (RST) provides important information about the relative importance of the different text spans in a document. This knowledge could be useful for sentiment analysis and polarity classification. However, RST has only been studied for polarity classification problems in constrained and small scale scenarios. The main objective of this paper is to explore the usefulness of RST in large-scale polarity ranking of blog posts. We apply sentence-level methods to select the key sentences that convey the overall on-topic sentiment of a blog post. Then, we apply RST analysis to these core sentences in order to guide the classification of their polarity and thus to generate an overall estimation of the document's polarity with respect to a specific topic. Our results show that RST provides valuable information about the discourse structure of the texts that can be used to make a more accurate ranking of documents in terms of their estimated sentiment in multi-topic blogs.

Keywords: Blog, Opinion Mining, Sentiment Analysis, Polarity Estimation, Discourse Structure, Rhetorical Structure Theory.

1 Introduction

Social networks and blogs have rapidly emerged to become leading sources of opinions in the Web. These repositories of opinions have become one of the most effective ways to influence people's decisions. In fact, companies are aware of the power of social media and most enterprises try to monitor their reputation over Twitter, blogs, etc. to infer what people think about their products and to get early warnings about reputation issues. In this paper, we focus on one of the most important sources of opinions in social media, i.e., the blogosphere [1]. In this

scenario, classical information retrieval (IR) techniques are not enough to build an effective system that deals with the opinionated nature of these new sources of information. To mine opinions from blogs we need to design methodologies for detecting opinions and determining their polarity [2].

In recent years, several works have been conducted to detect opinions in blog posts [1]. Currently, the most popular approach is to consider this mining task as a two-stage process that involves a topic retrieval stage (i.e., retrieve relevant posts given a user query), and a re-ranking stage that takes into account opinion-based features [3]. This second stage can also be subdivided into two different substaks: an opinion-finding task, where the main aim is to find opinionated blog posts related to the query, and a subsequent polarity task to identify the orientation of a blog post with respect to the topic (e.g., positive or negative). For polarity estimation, researchers often apply naive methods (e.g., classifiers based on frequency of positive/negative terms) [4]. Polarity estimation is a really challenging task with many unresolved issues (e.g., irony, conflicting opinions, etc.). We argue that this difficult estimation problem cannot be solved with regular matching (or count-based) techniques alone. In fact, most lexicon-based polarity classification techniques fail to retrieve more positive/negative documents than baselines without polarity capabilities [3].

This phenomenon is caused by the polarity of a document being not so much conveyed by the sentiment-carrying words that people use, but rather by the way in which these words are used. Rhetorical roles of text segments and their relative importance should be accounted for when determining the overall sentiment of a text (e.g., an explanation may contribute differently to the overall sentiment than a contrasting text segment does) [5]. Rhetorical Structure Theory (RST) [6] is a linguistic method for describing natural text, characterizing its structure primarily in terms of relations that hold between parts of the text. Rhetorical relations (e.g., an explanation or a contrast) are very important for text understanding, because they give information about how the parts of a text are related to each other to form a coherent discourse.

Accounting for the rhetorical roles of text segments by means of a RST-based analysis has proven to be useful when classifying the overall document-level polarity of a limited set of movie reviews [5]. As this success comes at a cost of computational complexity, the application of a RST-based analysis in large-scale polarity ranking tasks in the field of IR is challenging. In this paper, we study how we can utilize RST in a large-scale polarity ranking task and how RST helps to understand the sentiment expressed by bloggers. More specifically, we aim to identify the rhetorical relations that give good guidance for understanding the sentiment conveyed by blog posts, as well as to quantify the advantage of exploiting these relations. We also compare our RST-based methods with conventional approaches for large-scale polarity ranking of blog posts.

In the blogosphere, the presence of spam, off-topic information, or *relevant* information that is non-opinionated introduces noise and this is a major issue that harms the effectiveness of opinion-finding techniques. Therefore, it would not be wise to apply RST on the entire blog posts. We build on recent advances in

extracting key opinionated sentences for polarity estimation in blog posts [4] and analyse the structure of the discourse only for selected passages. This is beneficial to avoid noisy chunks of text and it is also convenient from a computational complexity perspective because discourse processing is not lightweight.

2   Method

First, we present the methods to find *relevant* polar sentences in a blog post. Then, we show how to perform rhetorical analysis over these key evaluative sentences, in order to determine the relations between the different spans of text. Finally, we define the overall orientation of a blog post as positive (resp. negative) according to these key evaluative sentences. To this end, we take into account the information provided by rhetorical relations.

2.1   Finding Relevant Polar Sentences

Many efforts have been recently made to determine what are the important parts of a document for polarity purposes. Most authors [7,8,9] have studied this issue in a typical IR scenario, i.e., given a query, the system has to return a ranking of positive opinionated documents and a ranking of negative opinionated documents [3]. This task is approached as a re-ranking task in which systems first retrieve a list of relevant documents and then reorganize them according to their polarity. In this paper we follow the same approach.

We apply an effective and efficient approach [4] based on sentence retrieval and a well-know sentiment classifier, OpinionFinder (OF) [10]. OpinionFinder estimates what sentences are subjective and also marks various aspects of the subjectivity in the sentences, including the source (holder) of the opinions and the words that are included in phrases expressing positive or negative sentiments. The information provided by OF was very useful for both subjectivity and polarity estimation in numerous experimental validations [4,8,9,11].

Basically, the terms tagged by OF as positive or negative are used to define the positive or negative polarity score of a sentence. Furthermore, to promote polar sentences that are on-topic (i.e., sentences that are relevant to the query topic), sentence retrieval is applied to determine the relatedness between the query terms and each polar sentence. To this end, we use the Lemur[1] implementation of tf-idf, with BM25-like weights[2] as our sentence retrieval method. BM25 [12] is a robust and effective IR model that has shown its merits in many search tasks.

Finally, the combination of relevance and polarity is done through linear interpolation:

$$pol(S,Q) = \beta \cdot rel_{norm}(S,Q) + (1 - \beta) \cdot pol(S), \tag{1}$$

where Q is the query, $rel_{norm}(S,Q)$ is the Lemur's tf-idf score after a query-based normalization into [0,1] and $pol(S)$ represents the number of positive (resp.

---

[1] `http://www.lemurproject.org/`

[2] We build a sentence-level index and apply the well-known BM25 suggested configuration ($k1 = 1.2, b = 0.75$).

negative) terms tagged in the sentence S divided by the total number of terms in S^3. $\beta \in [0,1]$ is a free parameter.

Different aggregation methods were considered in [4] to compute the final polarity of a blog post based on its sentence-level scores, including the average score of all polar sentences, the first or the last k polar sentences and the sentences with the highest $pol(S,Q)$. This last method, *PolMeanBestN*, was shown to be very robust and, overall, it gives the best estimation of the polarity of a blog post. Therefore, in this paper, we use this approach to extract the key sentences that are injected to a RST module. The best configuration obtained in [4] for *PolMeanBestN* is $k = 1$, which means that we select just one sentence to estimate the overall polarity of a blog post.

Given an initial list of documents which is ranked by decreasing relevance score $(rel_{norm}(D,Q))$, we re-rank the list to promote on-topic blog posts that are positive (resp. negative) opinionated as follows:

$$pol(D,Q) = \gamma \cdot rel_{norm}(D,Q) + (1-\gamma) \cdot pol_S(D,Q), \qquad (2)$$

where rel_{norm} is the document's relevance score after a query-based normalization in [0,1], $pol_S(D,Q) = max_{S \in D} pol(S,Q)$ (e.g., *PolMeanBestN*, with k=1), and $\gamma \in [0,1]$ is a free parameter[4].

2.2   Rhetorical Structure Theory

Discourse analysis is concerned with how meaning is built up in the larger communicative process. Such an analysis can be applied on different levels of abstraction, i.e., within a sentence, within a paragraph, or – typically – within a document or conversation. The premise is that each part of a text has a specific role in conveying the message of a piece of natural language text. RST [6] is one of the leading discourse theories. The theory can be used to split texts into segments that are rhetorically related to one another. Each segment may in turn be split as well, thus yielding a hierarchical rhetorical structure. Within this structure, text segments can be either nuclei or satellites, with nuclei being assumed to be more significant than satellites with respect to understanding and interpreting a text. Many types of relations between text segments exist; the main paper on RST defines 23 types of relations [6]. A satellite may for instance be an elaboration on what is explained in a nucleus. It can also form a contrast with respect to matters presented in a nucleus.

For an example of a RST-structured sentence, let us consider the sentence "*Although I like the characters, the book is horrible.*", which can be split into two segments. The core of the sentence, i.e., the nucleus, provides a negative

---

[3] For positive document retrieval $pol(S)$ is the percentage of positive terms in the sentence, and for negative document retrieval $pol(S)$ is the ratio of negative terms in the sentence.

[4] We used the configuration provided in [4] for the parameters β and γ ($\beta = 0.6, \gamma = 0.6$ for negative polarity estimation, and $\beta = 0.2, \gamma = 0.5$ for positive polarity estimation). This configuration was shown to be very stable across different collections.

sentiment with respect to a book ("*the book is horrible*"). The other segment is a satellite with contrasting information with respect to the nucleus, admitting to some positive aspects of the book ("*Although I like the characters*"). For a human reader, the polarity of this sentence is clearly negative, as the overall message has a negative polarity. However, in a classical (word-counting) sentiment analysis approach, all words would contribute equally to the total sentiment, thus yielding a verdict of a neutral or mixed polarity at best. Exploiting the information contained in the RST structure could result in the nucleus being given a higher weight than the satellite, thus shifting focus to the nucleus segment. We can thus get a more reliable sentiment score. As such, in order to exploit the rhetorical relations as imposed upon natural language text by a RST analysis, distinct rhetorical roles of individual text segments should be treated differently when aggregating the sentiment conveyed by these text segments. This could be accomplished by assigning different weights to distinct rhetorical roles, quantifying their contribution to the overall sentiment conveyed by a text [5].

2.3 Sentence-Level Parsing of Discourse

In order to automatically structure our identified key evaluative sentences by means of a RST-based analysis, we used SPADE (Sentence-level PArsing of DiscoursE)[13], which creates RST trees for individual sentences. SPADE was trained and tested on the train and test set of the RST Discourse Treebank (RST-DT) [14], achieving a F1 score of 83.1% on identifying the right rhetorical relations and their right arguments [13]. The relations taken into account in our experiments are detailed in Table 1.

2.4 RST over On-topic Polar Sentences

To include RST in our method, we compute $pol(S)$ as a weighted sum of the polar terms occurring in the nucleus and the satellite, respectively:

$$pol(S) = w_{nuc} \cdot pol_{nuc}(S) + w_{sat} \cdot pol_{sat}(S), \qquad (3)$$

where nuc represents the nucleus of the sentence S, sat is the satellite of the sentence S, w_{nuc} is the weight for nucleus, w_{sat} is the weight for the concrete satellite and $pol_{nuc}(S)$ and $pol_{sat}(S)$ represent the ratio of positive (resp. negative) terms tagged in the nucleus and satellite respectively of sentence S. Observe that w_{sat} and w_{nuc} are free parameters that need to be trained for each different rhetorical relation. Finally, observe that despite the fact that RST is a computationally intensive task[5], this process can be done offline (at indexing time).

3 Experiments

In this section, we describe the experiments designed to determine the usefulness of RST in a large-scale multi-topic domain. Concretely, we work with the

---

[5] SPADE software takes on average 3 seconds to compute each sentence in a regular desktop machine.

Table 1. RST relation types taken into account

Relation	Description
attribution	Clauses containing reporting verbs or cognitive predicates related to reported messages presented in nuclei.
background	Information helping a reader to sufficiently comprehend matters presented in nuclei.
cause	An event leading to a result presented in the nucleus.
comparison	Clauses presenting matters which are examined along with matters presented in nuclei in order to establish similarities and dissimilarities.
condition	Hypothetical, future, or otherwise unrealized situations, the realization of which influences the realization of nucleus matters.
consequence	Information on the effects of events presented in nuclei.
contrast	Situations juxtaposed to situations in nuclei, where juxtaposed situations are considered as the same in many respects, yet differing in a few respects, and compared with respect to one or more differences.
elaboration	Rhetorical elements containing additional detail about matters presented in nuclei.
enablement	Rhetorical elements containing information increasing a readers potential ability of performing actions presented in nuclei.
evaluation	An evaluative comment about the situation presented in the associated nucleus.
explanation	Justifications or reasons for situations presented in nuclei.
joint	No specific relation is assumed to hold with the matters presented in the associated nucleus.
otherwise	A situation of which the realization is prevented by the realization of the situation presented in the nucleus.
temporal	Clauses describing events with a specific ordering in time with respect to events described in nuclei.

BLOGS06 text collection [15], which is one of the most renowned blog test collections with relevance, subjectivity, and polarity assessments.

3.1 Collection and Topics

We take into account the TREC 2006, TREC 2007, and TREC 2008 blog track's benchmarks. All these tracks have the BLOGS06 as the reference collection for experiments. Each year, a new set of 50 topics was provided and new judgments were made according to the documents retrieved by the participants. One of the core tasks of these tracks is the polarity task, i.e., given a query topic, systems have to return a ranking of positive (resp. negative) blog posts related to the query. Each query topic contains three different fields (i.e., title, description, and narrative). In this work we only utilise the title field, which is short and the best representation of real user web's queries, as reflected in the official TREC Blog track literature [3]. Documents and topics are pre-processed with Krovetz stemmer and we remove 733 English stopwords.

Documents were judged by TREC assessors in two different aspects: i) Topic relevance: a post can be relevant, not relevant, or not judged, ii) Opinion: whether the on-topic documents contain explicit expression of opinion or sentiment about the topic then the document is tagged as positive, negative, or mixed (if the opinion expressed is ambiguous, mixed, or unclear).

3.2   Retrieval and Polarity Baselines

In TREC 2008, to allow the study of the performance of a specific opinion-finding technique across a range of different topic-relevance baseline systems, a set of five topic-relevance baselines was provided. These standard baselines use a variety of different retrieval approaches, and have varying retrieval effectiveness[6].

Spam detection, topic retrieval in blogs, and subjectivity classification are out of the scope of this paper. We focus on the effect of RST on the set of subjective documents identified by the standard baseline runs. This means that the input to our methods is a set of opinionated documents with varied polarity orientations (positive, negative, or mixed polarity) and the objective is to distinguish the type of polarity that every document has (i.e., search for positive, and search for negative documents). This polarity task, per se, is quite challenging because there are many offtopic passages and conflicting opinions. The measures applied to evaluate performance are mean average precision (MAP), and precision at 10 documents (P@10). These measures are commonly applied to assess the performance of ranking algorithms.

3.3   Training and Testing

We have built a realistic and chronologically organised query dataset with the topics provided by TREC. We have optimised the parameters of our methods (e.g., satellite weights) on the TREC 2006 and TREC 2007 topics, while using the TREC 2008 topics as testing set. Two different training-testing processes focused on maximising MAP have been run, i.e., one for positive polarity retrieval and another for negative polarity retrieval. To train all the parameters of our models (including the satellite weights) we have used Particle Swarm Optimisation (PSO). PSO has shown its merits for the automatic tuning process of the parameters of IR methods [16].

3.4   Results

Table 2 shows the results of our polarity approaches. Each run is evaluated in terms of its ability to retrieve positive (resp. negative) documents higher up in the ranking. The best value in each column for each baseline is underlined. Statistical significance is assessed using the paired t-test at the 95% level. The symbols ▲ and ▼ indicate a significant improvement or decrease over the corresponding baseline. To specifically measure the benefits of RST techniques in the estimation of a ranking of positive (resp. negative) blog posts we compare its performance against the performance achieved by a very effective method for blog polarity estimation (*PolMeanBestN* [4], presented in Section 2). *PolMeanBestN* estimates the overall recommendation of a blog post by taking into account the on-topic sentence in the blog post that has the highest polarity score (e.g., the

---

[6] The baselines were selected by TREC from the runs submitted to the initial ad-hoc retrieval task in the TREC blog track.

Table 2. Polarity Results. Mean average precision (MAP) and precision at 10 (P10) for positive and negative rankings of blog posts. The symbols $\triangle$ ($\triangledown$) and $\blacktriangle$($\blacktriangledown$) indicate a significant improvement(decrease) over the original baselines provided by TREC and the *polMeanBestN* method, respectively.

	Negative		positive	
	MAP	P10	MAP	P10
baseline1	.2402	.2960	.2662	.3680
+polMeanBestN	.2408	.3000	.2698	.3720
+polMeanBestN(RST)	.2516	.3180△ ▲	.2733	.3740△ ▲
baseline2	.2165	.2780	.2390	.3340
+polMeanBestN	.2222	.2820	.2368	.3160
+polMeanBestN(RST)	.2261▲	.3100△ ▲	.2423△	.3560△ ▲
baseline3	.2488	.2840	.2758	.3500
+polMeanBest	.2524	.2760	.2755	.3420
+polMeanBestN(RST)	.2584△ ▲	.2820	.2770△	.3380▼
baseline4	.2636	.2740	.2731	.3580
+polMeanBestN	.2730	.2840	.2705	.3500
+polMeanBestN(RST)	.2825△	.3240△ ▲	.2716	.3620△ ▲
baseline5	.2238	.3000	.2390	.3600
+polMeanBestN	.2279	.3120	.2404	.3580
+polMeanBestN(RST)	.2393	.3420△ ▲	.2786△ ▲	.4380△ ▲

most controversial contents of the post). This configuration leads to a performance comparable to the best performing approach at the TREC 2008 Blog track (KLE system) [1,4]. Observe that the RST technique proposed in our paper is an evolution over *PolMeanBestN*, in which the estimation of polarity is also done with the highest polarity sentence but we take into account its RST structure (eq. 2). The symbols $\triangle$ and $\triangledown$ indicate a significant improvement or decrease over this polarity method.

Polarity retrieval performance. The technique that performs the best across all different baselines is the RST-based method, showing usually significant improvements with respect to both the baseline and *PolMeanBestN*. Another important finding is that *PolMeanBestN* never significantly outperforms the baselines.

Positive vs Negative results. Another observation is that the performance of negative document rankings is lower than the performance of positive document rankings. This may be caused by negative documents being harder to find. As a matter of fact, there are more positive documents than negative ones in the polarity judgements (3,338 against 2,789). Additionally, the lexicon-based identification of negative documents may be thwarted by people having a tendency of using rather positive words to express negative opinions [5].

Optimised weights for relations. Table 3 shows the weights learnt for the different RST elements. The weight of the nucleus was fixed to one. Weights of satellites are real numbers in the interval $[-2, 2]$. Having been assigned a weight of 1, nuclei are assumed to play a more or less important role in conveying the overall sentiment of a piece of natural language text. Yet, some types of satellites appear to play an important role as well in conveying the overall sentiment of a document. For instance, the most salient relations (highest percentage of appearance in the collection) in our training set appear to be the *elaboration*

Table 3. Optimised weights for RST relation types trained with PSO over positive and negative rankings and the percentage of presence of different relations in the training

Relation	Positive		Negative	
	% of Presence	Weight	% of Presence	Weight
attribution	.183	0.531	.177	2.000
background	.034	-0.219	.038	-2.000
cause	.009	1.218	.009	-0.011
comparison	.003	-1.219	.003	-2.000
condition	.029	-0.886	.025	-2.000
consequence	.001	0.846	.001	1.530
contrast	.016	-1.232	.017	-2.000
elaboration	.207	2.000	.219	2.000
enablement	.038	2.000	.038	1.221
evaluation	.001	0.939	.001	-2.000
explanation	.007	2.000	.008	2.000
joint	.009	-1.583	.010	1.880
otherwise	.001	-1.494	.001	-0.428
temporal	.003	-2.000	.003	-0.448

and the *attribution* relation. For both positive and negative documents, satellite segments elaborating on matters presented in nuclei are typically assigned relatively high weights, exceeding those assigned to nuclei. Bloggers may, therefore, tend to express their sentiment in a more apparent fashion in elaborating segments rather than in the core of the text itself. A similar pattern emerges for attributing satellites as well as for persuasive text segments, i.e., those involved in *enablement* relations, albeit to a more limited extent (lower frequency of occurrence). Interestingly, however, the information in attributing satellites appears to be more important in negative documents than in positive documents. Another important observation is that the sentiment conveyed by elements in contrast satellites gets a negative weight. This permits to appropriately estimate the polarity of sentences such as the one we introduced in Section 2 (*"Although I like the characters, the book is horrible."*).

4   Related Work

Numerous studies have been conducted to determine opinions in blog posts. In large-scale scenarios the search for subjective documents (regardless of their polarity) has been studied in detail [7,8,9]. Most successful studies in this area try to find document that are both opinionated and on-topic [7,8]. To perform this task, some authors consider positional information as the best guidance to find opinions related to the query. For example, Santos et al. [8] used the proximity of query terms to subjective sentences in a document to detect on-topic opinions. In a similar way, Gerani et al. [7] proposed a proximity-based opinion propagation method to calculate the aggregated opinion at the position of each query term in a document.

Pang and Lee [17] considered the use of the location of the opinionated sentences on the accuracy of two state-of-the art polarity classifiers of film reviews. They built polarity classifiers based on sentences from different parts of a document (e.g. first sentences, last sentences), however these classifiers were not able

to overcome local-unigram state-of-the-art systems. Nevertheless, the results obtained showed that the last sentences of a document might be a good indicator of the overall polarity of the review.

In [18], Zirn et. al. presented an automatic framework for fine-grained sentiment analysis at sub-sentence level in a product review scenario. They combined several sentiment lexicons with neighborhood information and discourse relations to enhance polarity performance. Concretely, they used Markov logic to integrate polarity scores from different sentiment lexicons with information about relations between neighboring segments of texts. They demonstrated that the use of structural features improves the accuracy of polarity predictions achieving accuracy scores of up to 69%. In our paper we have studied the impact of structural information in a more demanding multi-topic scenario, the blogosphere.

Somasundaran et al. [19] demonstrated the importance of general discourse analysis in polarity classification of multi-party meetings. The importance of RST for the classification of ambiguous sentences (i.e., sentences with conflicting opinions) was studied in [20]. Closer to our work, Heerschop et. al. [5] worked with film reviews and used RST to determine the importance of every piece of text in the review for polarity classification. By dividing the text into important and less important parts, depending on their rhetorical role according to a sentence-level RST-analysis, they were able to outperform a whole-document approach based on polarity lexicons. One of the main issues that the authors found in their experiments was the processing time required for identifying/classifying discourse structure in natural language text. This problem prevents the application of these methods in large-scale scenarios. In our work we have revisited this issue and we have studied and successfully applied rhetorical relations in a large-scale scenario.

In [21], Lioma et. al. designed a Language Model (LM) that takes into account RST information to estimate the relevance of a document to a query in a web search scenario. Their experiments showed that some rhetorical relations lead to important gains in performance over state-of-the-art retrieval methods.

Chenlo and Losada [4] proposed some effective and efficient methods to find the opinionated passages of a blog post that are on-topic. By combining simple sentence retrieval methods and polarity evidence, the authors were able to represent the overall opinion of a blog post by selecting just a few sentences from the beginning, from the end or from the set of most subjective and on-topic sentences of the document. In our current endeavours, we have used this method to focus our RST analysis on the core parts of documents and also to avoid the problems related to the use of a computationally expensive method such as RST-based polarity analysis.

5   Conclusions and Future Work

In this paper we have taken the first steps towards studying the usefulness of RST-based polarity analysis in the blogosphere. We found that the use of discourse structure significantly improves polarity detection in blogs. We have

applied an effective and efficient strategy to select and analyse key opinion sentences in a blog post and we have found some trends related to the way in which people express their opinions in blogs. Concretely, there is a clear predominance of attribution and elaboration rhetorical relations. Bloggers tend to express their sentiment in a more apparent fashion in elaborating and attributing text segments rather than in the core of the text itself.

Finally, most of the methods proposed on this work are based on a simple combination of scores. As future work, we would like to study more formal combination methods. Related to this, we are also interested in more refined representations of rhetorical relations (e.g., LMs [21]). Another problem to take into account is that we are using only one sentence to evaluate the polarity of the blog post. Under these conditions the benefits of applying rhetorical relations have some limitations (e.g., the sentence selected may not be a good representative for the blog post). In the near future, we plan to explore the benefits of discourse structure while taking more sentences into account in our analysis. Related to this, one of the core problems derived to the use of RST is the processing time required for identifying discourse structure in natural language text. Therefore, we would like to explore more efficient methods of identifying the discourse structure of texts.

References

1. Santos, R.L.T., Macdonald, C., McCreadie, R., Ounis, I., Soboroff, I.: Information retrieval on the blogosphere. Found. Trends Inf. Retr. 6(1), 1–125 (2012)
2. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2007)
3. Ounis, I., Macdonald, C., Soboroff, I.: Overview of the TREC 2008 blog track. In: Proc. of the 17th Text Retrieval Conference, TREC 2008. NIST (2008)
4. Chenlo, J.M., Losada, D.: Effective and efficient polarity estimation in blogs based on sentence-level evidence. In: Proc. 20th ACM Int. Conf. on Information and Knowledge Management, CIKM 2011, Glasgow, UK, pp. 365–374 (2011)
5. Heerschop, B., Goossen, F., Hogenboom, A., Frasincar, F., Kaymak, U., de Jong, F.: Polarity analysis of texts using discourse structure. In: Proc. 20th ACM Int. Conf. on Inf. and Knowledge Manag., CIKM 2011, Glasgow, UK, pp. 1061–1070 (2011)
6. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text 8(3), 243–281 (1988)
7. Gerani, S., Carman, M.J., Crestani, F.: Proximity-based opinion retrieval. In: Proc. 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, pp. 403–410. ACM, New York (2010)

 8. Santos, R.L.T., He, B., Macdonald, C., Ounis, I.: Integrating proximity to subjective sentences for blog opinion retrieval. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 325–336. Springer, Heidelberg (2009)
 9. He, B., Macdonald, C., He, J., Ounis, I.: An effective statistical approach to blog post opinion retrieval. In: Proc. 17th ACM Int. Conf. on Information and Knowledge Management, CIKM 2008, pp. 1063–1072. ACM, New York (2008)
10. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proc. Conf. on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005, pp. 347–354. ACL (2005)
11. He, B., Macdonald, C., Ounis, I.: Ranking opinionated blog posts using opinion-finder. In: SIGIR, pp. 727–728 (2008)
12. Robertson, S.: How okapi came to TREC. In: Voorhees, E.M., Harman, D.K. (eds.) TREC: Experiments and Evaluation in Information Retrieval, pp. 287–299 (2005)
13. Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: Proc. 2003 Conf. of the North American Chapter of the ACL on Human Language Technology, NAACL 2003, vol. 1, pp. 149–156. ACL, Stroudsburg (2003)
14. Carlson, L., Marcu, D., Okurowski, M.E.: Building a discourse-tagged corpus in the framework of rhetorical structure theory. In: Proc. 2nd SIGdial Workshop on Discourse and Dialogue, SIGDIAL 2001, vol. 16, pp. 1–10. ACL (2001)
15. Macdonald, C., Ounis, I.: The TREC Blogs 2006 collection: Creating and analysing a blog test collection. Technical Report TR-2006-224, Department of Computing Science, University of Glasgow (2006)
16. Parapar, J., Vidal, M., Santos, J.: Finding the best parameter setting: Particle swarm optimisation. In: 2nd Spanish Conf. on IR, CERI 2012, pp. 49–60 (2012)
17. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Pr. of the ACL, pp. 271–278 (2004)
18. Zirn, C., Niepert, M., Stuckenschmidt, H., Strube, M.: Fine-grained sentiment analysis with structural features. In: Asian Federation of Natural Language Processing, vol. 12 (2011)
19. Somasundaran, S., Namata, G., Wiebe, J., Getoor, L.: Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In: Proc. 2009 Conf. on Empirical Methods in Natural Language Processing, EMNLP 2009, vol. 1, pp. 170–179. ACL (2009)
20. Zhou, L., Li, B., Gao, W., Wei, Z., Wong, K.F.: Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In: Proc. Conf. on Empirical Methods in Natural Language Processing, EMNLP 2011, pp. 162–171. ACL, Stroudsburg (2011)
21. Lioma, C., Larsen, B., Lu, W.: Rhetorical relations for information retrieval. In: Proc. 35th Int. Conf. ACM SIGIR on Research and Development in Information Retrieval, SIGIR 2012, pp. 931–940. ACM, New York (2012)

Automatic Detection of Ambiguous Terminology for Software Requirements

Yue Wang, Irene L. Manotas Gutiérrez, Kristina Winbladh, and Hui Fang

Department of Electrical and Computer Engineering,
University of Delaware,
Newark, DE 19716
{wangyue,imanotas,winbladh,hfang}@udel.edu

Abstract. Identifying ambiguous requirements is an important aspect of software development, as it prevents design and implementation errors that are costly to correct. Unfortunately, few efforts have been made to automatically solve the problem. In this paper, we study the problem of lexical ambiguity detection and propose methods that can automatically identify potentially ambiguous concepts in software requirement specifications. Specifically, we focus on two types of lexical ambiguities, i.e., *Overloaded* and *Synonymous* ambiguity. Experiment results over four real-world software requirement collections show that the proposed methods are effective in detecting ambiguous terminology.

Keywords: Ambiguity detection, Software requirements, Overloaded ambiguity, Synonymous ambiguity.

1   Introduction

A Software Requirements Specification (SRS) describes the required behaviour of a software product, and is often specified as a set of necessary requirements for project development. An ideal SRS should clearly state the requirements without introducing any ambiguities. Unfortunately, it is impossible to avoid the ambiguous SRSs since they are often described using natural languages.

A requirement is ambiguous if it can be interpreted in multiple ways. Ambiguous requirements can be a major problem in software development [4]. Project participants tend to subconsciously disambiguate requirements based on their own understanding without realizing that they are ambiguous. As a result, different interpretations often remain undiscovered until later stages of the software life-cycle, when design and implementation choices materialize the specific interpretations. It costs 50-200 times as much to correct an error late in a software project compared to when it was introduced [3].

One possible way of preventing ambiguous requirements is through manual inspection [17], which clearly is time-consuming and error prone. Consequently, it is important to study how to automatically detect ambiguous requirements in software requirement specifications (SRS).

Establishing a consistent usage of terminology early on in a project is imperative as it provides a vocabulary for the project and can greatly reduce misunderstandings.

In this paper, we focus on the problem of lexical ambiguity detection. Specifically, we aim to detect terminology misuse such as overloaded and synonymous concepts. We use the word *concept* instead of *term*, because we consider both terms and phrases. A concept is *overloaded* if it refers to different semantic meanings and it is *synonymous* if several different concepts are used interchangeably to refer to the same semantic meaning (see Fig. 1). Note that overloaded concepts include both homonyms and polysemy.
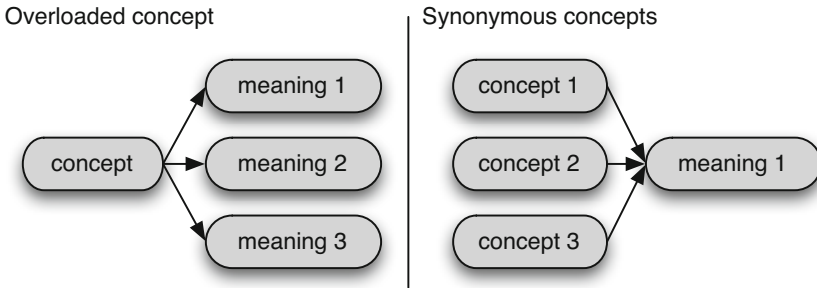
Fig. 1. Overloaded and synonymous concepts

We propose to formulate the problem as a ranking problem that ranks all the important concepts from a SRS based on their ambiguity scores. The ranked list of concepts is expected to help requirement engineers to more efficiently identify ambiguous concepts and revise the SRS accordingly. One advantage of formulating the problem this way is to allow requirements engineers to decide how many concepts they want to go through based on their own situations. For example, some engineers may want to catch all ambiguous concepts while others may only have limited time to correct the most ambiguous ones. Once the ambiguous concepts are identified and rephrased, the SRS would have higher quality and can be better used in the subsequent stages of the project.

Specifically, we propose two feature-based methods that can rank the concepts based on their overloaded and synonymous ambiguities respectively. Experiments are conducted over four data sets with real-world SRSs. These data sets cover different types and scales of software systems. Results show that the proposed methods are effective in detecting both overloaded concepts and synonyms.

2   Related Work

Requirements ambiguities can be avoided by using formal languages to specify the requirements. Formal languages use mathematical notations and syntax to specify requirements precisely and can be used to check the requirements for inconsistencies and other problems. A non-extensive list of formal approaches include approaches that use logic-based, state-based, event-based, and algebraic-based representations [5,6,8,22]. Although formal specification languages do avoid ambiguities, there are some limitations in using them. One limitation is that formal notations require more efforts from

requirements engineers and other participants in creating and reviewing requirements. Another limitation is that although a formally specified requirement might be free of ambiguities, it could still be incorrect as it has been translated from an informal requirement at some point. That is, the same disambiguating assumptions can be made when translating informal requirements into the formal notation as when leaving the requirements in their informal representation and using them in subsequent development activities. It is therefore important to disambiguate the language used in the informal representation prior to using a formal notation.

A common approach to handle ambiguous requirements problem in SRSs is the use of a project glossary. The creation of a project glossary generally occurs during domain understanding and requirements elicitation. Although a project glossary can play an essential role in a software project, there is usually no quality checks on the glossary. It turns out that many glossaries are rather weak in the sense that they do not cover the terminology that is actually used in a specification and the synonymous and overloaded concepts are not recognized and marked [25]. Chantree et al. present an interesting approach with a focus on identifying ambiguities that are likely to lead to misunderstandings [4]. Others have worked on resolving requirements ambiguities that are likely problematic to requirements engineers [2,18]. Our work differs in that we focus on terminology consistency and specifically on reducing the ambiguity that can result from terminology misuse.

Some studies tried to combine machine learning and NLP techniques to identify ambiguous requirements at the sentence level [10,15] . On the contrary, this paper focused on detecting the ambiguous requirements from at the concept level.

Our work is closely related to the word sense disambiguation problem, which determines the appropriate sense of a word given its context and the senses often are defined in a dictionary. However, our work focuses on a different problem, and aims to detect whether a concept is ambiguous in a requirements document. The problem is more challenging than in the general domain since the definition of the ambiguity is more subtle. First, the problem is domain-specific, and there is no dictionary available for each domain to describe possible senses of every concept, which requires us to automatically identify possible senses by ourselves. Second, the definition of ambiguity in SRSs is not well defined, and relies highly on the context of the concepts. A concept may be used ambiguously in the requirements of one project, but not in other projects.

3   Problem Formulation

A SRS is ambiguous if it can be interpreted in more than one ways [2]. There are many different types of ambiguities, and here we focus on lexical ambiguities. Lexical ambiguities can be classified into *overloaded ambiguity* and *synonymous ambiguity* [25], as shown in Figure 1. We define an overloaded ambiguity to be a concept that has lost its specificity in the particular document. For example, consider the concepts *user*, *guest user*, and *verified user* in a SRS. In cases where only *user* is used in the SRS, a reader may not be able to distinguish which kind of user is intended. In contrast to overloaded ambiguity, synonymous ambiguity is when multiple concepts refer to the same semantic meaning. For instance, in the SRS of a testing gateway system, the concepts *system* and

testing gateway both refer to the system to be developed. As a result, requirement engineers could use both concepts in the SRS without realizing the potential for conflicts and misunderstandings.

To detect ambiguous concepts from a SRS collection, we first use C-value method [24,7] to extract candidate concepts, and then propose to rank the extracted concepts or concept pairs based on their degree of ambiguity. In particular, for overloaded ambiguity detection, concepts should be ranked based on the likelihood that a concept has multiple interpretations, while for synonymous ambiguity detection, concept pairs are ranked based on the likelihood that they represent the same meaning. The ranked lists are expected to help requirements engineers focus on the concepts that are most likely to be ambiguous so that they can quickly identify the places that need clarification.

The key challenge here is how to estimate the ambiguity score for a concept or a concept pair. We focus on identifying useful features that could be used to identify each type of ambiguities. For overloaded ambiguities, the features are mostly related to the *context* of a concept, i.e., words that occur before and after the concept in the same sentence. For synonymous ambiguity detection, the features are based on not only context but also patterns and content of the candidate pairs. With the identified features, we then propose a possible solution to combine them and learn the ambiguity scores for the concepts or concept pairs. Details are provided in the following sections.

4   Overloaded Ambiguity Based Ranking

As defined previously, overloaded ambiguities lead to a "one-to-many" mapping from concepts to semantic meanings. Since the context of a concept is closely related to its semantic meaning, the degree of ambiguity of a concept should be determined by how diverse its context is. We study the following features that measure the diversity of the context for a concept.

- **Concept Frequency:** Given a concept, this feature computes the frequency of the concept in all the SRSs. The intuition is that a concept is more likely to cause an overload ambiguity when it occurs more frequently in the collection.
- **Context Diversity:** For a given concept, the feature measures how diversified its contexts are. We define a context of a concept as a set of words that occur in the same sentence as the concept. If the concept is overloaded, its context should cover different meanings for the sub-layer entities. Therefore, the diversity score should be high. On the other hand, the entity that the concept refers to should be consistent among different contexts, which means the context diversity should be low. The context diversity score of a concept is computed as the inverse of the average cosine similarity among all its contexts.
- **Number of Clusters in the Context:** Clustering is one possible way of partitioning contexts of a concept into different groups with similar meanings. Thus, the number of clusters could be a good indicator of the degree of ambiguity of the concept. In this paper, we use hierarchical agglomerative clustering method [12]. There are multiple ways for clustering. During the training stage of our experiment, we tried single-link, complete-link and centroid HAC algorithm. The results suggested that

the single link algorithm consistently outperform than the other algorithms. Therefore, it is chosen as the final method. We keep grouping similar contexts together until it reaches the stopping criterion, i.e., when the minimum similarity between each group is smaller than a *similarity boundary*.

– **Inter-Cluster Distance:** It measures the average distance among different clusters. The intuition is that when a concept is ambiguous, its context clusters would cover different information, which leads to higher inter-cluster distance. The distance is computed as the inverse of the similarity, which can be computed using cosine similarity based on the context.

We now discuss how to combine all the features. Since each feature can be used individually to rank concepts, we can then compute the ambiguity score of a concept based on its ranking positions using each of the features. The concepts are then ranked based on these scores.

Formally, c denotes a concept, $AS_O(c)$ denotes the overloaded ambiguity score of the concept, and $f_i(c)$ is the value of feature f_i for concept c. We can then have:

$$AS_O(c) = \sum \alpha_i \cdot PS(f_i(c))$$

where α_i is the weight of the result of each feature f_i and $\sum \alpha_i = 1$. The weights can be learned from a training set. $PS(x)$ is the relative position score of each feature and can be computed as:

$$PS(f_i(c)) = 1 - \frac{PositionInFeature(c, f_i) - 1}{\#TotalConcepts} \tag{1}$$

where the PositionInFeature is the ranking of concept c in feature f.

Note that there could be other ways of combining these features. We choose to use the relative value instead of the absolute score from each feature is because we want to make the results from different features more comparable.

5   Synonymous Ambiguity Based Ranking

A synonymous ambiguity is caused by a "many-to-one" mapping between concepts and semantic meanings. We identify the following features that can be used to identify synonymous ambiguity:

– **Context-Based Similarity:** It computes the average similarity of contexts for each pair of concepts. However, it is possible that two concepts have similar contexts but are not synonymous. For example, concepts *user ID* and *password* may co-occur frequently in a SRS collection, but they are not considered as synonymous. To solve this problem, we propose to consider only concept pairs that do not occur in the same sentence when computing the context similarities. Thus, the context-based similarity of two concepts c_i and c_j can be computed as follows:

$$Sim_C(c_i, c_j) = \frac{\sum_{x \in UC(c_i|c_j), y \in UC(c_j|c_i)} Similarity(x, y)}{|UC(c_i|c_j)| \times |UC(c_j|c_i)|}$$

where $UC(c_i|c_j)$ is a set of context for concept c_i that do not contain concept c_j. $Similarity(x, y)$ measures the similarity between two contexts and is computed using cosine similarity.

– **Pattern-Based Similarity:** Pattern-based features have been used to detect the semantic relationship in large text corpora [9,19,14]. We follow a similar strategy to detect synonym pairs in this paper. In particular, we start with a set of known pairs of synonymous concepts, and then retrieve the sentences that mention both concepts. We then identify patterns, i.e., common phrases or terms, and these patterns will then use to retrieve more candidate pairs. The process is repeated until no more new patterns can be found.
If concept c_i and c_j follow the discovered pattern P, then we have

$$Sim_P(c_i, c_j) = Sim_P(c_j, c_i) = 1.$$

Following the proposed methods, we are able to find the following patterns:
- c_1 **abbreviated** c_2
- c_1 **(c_2)**
- c_1**, also known as** c_2
- c_1**, a.k.a.** c_2

– **Textual-Based Similarity:** A synonymous concept pair reflects the same semantic meaning, so it is likely that their textual similarity is higher than other pairs. For example, concepts *account reference number* and *original account number* both refer to the number assigned to a user when opening an account. Thus, we have

$$Sim_T(c_i, c_j) = CosineSimilarity(c_i, c_j).$$

Each of the features captures one aspect of the synonymous ambiguities, and they all have their own limitations. Context-based similarity feature may fail to detect the ambiguous pairs from the same sentence, while pattern-based feature can mainly detect those from the same sentence. Textural similarity is only effective when the ambiguous pairs share common terms, and would fail to detect many that do not satisfy the requirement (e.g., the *account reference number* and its abbreviation *arn*). Thus, we propose the following method to combine all the features to improve the performance:

$$AS_S(c_i, c_j) = max\{Sim_P(c_i, c_j), (\alpha \cdot PS(Sim_C(c_i, c_j)) + (1 - \alpha) \cdot PS(Sim_T(c_i, c_j)))\}$$

where $PS(x)$ is the relative position score as shown in Equation (1). The proposed method trusts the results of pattern-based similarity more than other two features. When the two concepts do not follow any learned patterns, we will the consider their context and textual similarities. The importance between these two similarities is determined by the parameter α.

6   Experiment Setup

6.1   Experiment Design

Our system takes a set of SRSs as input, and then returns two separate ranking lists for the two kinds of ambiguities.

The pre-processing of the SRSs is kept to the minimum. We split the requirements into sentences, but did not remove stop words or stem the words. Stop words are not

Table 1. Description of data sets

	Type	Domain	SRS length	# of Req	Req Length	# of Rev.
PI	Web-based software engineering tool	Software Engineering	7524	62	14	7
PII	Web-based business application	Business	5711	65	17	11
PIII	Web-based lending application	Banking	26823	272	14	16
PIV	Business application	Business	2294	60	29	17

removed because they may be considered a stop word in one part of the document but used in a meaningful way in other parts of the document. For example, the words *to, be* are generally considered as stop words, but if these two words are removed, the concept *system to be* will lose its meaning. Word stemming is not used here because it may generate new ambiguity. For example, the concepts *programs, programmer, programming* are used correctly in the document without ambiguity. If word stemming is used, the three concepts will change to *program*, which could unnecessarily make the problem of overloaded ambiguity more difficult.

Results are evaluated with three measures, i.e., P@N (i.e., precision at top N results), R@N (i.e., recall at top N) and MAP@N (i.e., mean average precision at top N). P@N measures the percentage of top N detected concepts (or concept pairs) that are indeed ambiguous. R@N measures the percentage of ambiguous concept (or concept pairs) that are included in the top N results. MAP@N is a commonly used measure to evaluate the ranking results of top N results. Our primary evaluation measure is MAP@10.

6.2 Data Sets

We conduct experiments over four real-world data sets obtained from different software projects. These projects are chosen because they are real-world software projects, they span different domains and sizes, and there have been consistent efforts on revising the requirement documents. The characteristics of these projects are described in Table 1. The information includes the project name, project type, project domain, SRS length (in Terms), number of requirements, average requirement length and the number of revisions to the requirement documents for each project. The participants involved in **PI** and **PII** were software engineering students and professional developers with varying skills and experience, while those for **PIII** and **PIV** were professional developers.

To quantitatively evaluate the proposed approach, we create judgments on both ambiguity types for each project. Each judgment indicates whether a concept is overloaded ambiguous or whether a concept pair is synonymous ambiguous. The judgments are created by five assessors with training in software engineering and requirement engineering. For overloaded ambiguity, an assessor would go over all the candidate concepts for a project, and then decide whether each of them is ambiguous or not. The decision is made by first locating all the places where the concept was mentioned, and then check whether the concept has multiple meanings by reading the context of the concepts. The process for synonymous ambiguity is similar, while the assessor needs to compare the contexts of concept pairs.

The four projects were cross-evaluated by different assessors, for each project, there are at least 3 judgments for each type of ambiguity. With this judgments file, a voting schema is used to make the final decision. For each type of ambiguity of each project, we consider the candidate concept (pair of concepts) as ambiguous only if two or more assessors identified it is ambiguous.

Table 2 describes the basic statistics of the created judgments for each project. It includes the number of candidate concepts (i.e., *Concepts*), the number of overloaded concepts (i.e., *Overloaded*) and the number of synonymous concept pairs (i.e., *Synonymous*). It is surprising to see that a significant portion of the candidate concepts are still ambiguous even after at least 7 revisions, which reinforces the need for automated techniques that can help reduce these ambiguities and produce more consistent SRSs.

Table 2. Statistics of judgment sets

Projects	Concepts	Overloaded	Synonymous
PI	80	8	9
PII	66	23	3
PIII	143	11	7
PIV	57	7	6

7   Experiment Results

We now report the results for the proposed methods. There are several parameters in the proposed methods, so we train the parameter values on one collection (i.e., **PI**) and use the learned parameters for the remaining three test collections (i.e. **PII**, **PIII** and **PIV**). We conduct two sets of experiments to evaluate the effectiveness of the proposed methods for each ambiguity type, and report the optimal performance on the training set and the test performance on the testing sets for both sets.

7.1   Effectiveness of Overloaded Ambiguity Detection

Table 3 shows the optimal performance of the proposed overloaded ambiguity detection methods for **PI**. **All** denotes the method that combines all the features. **CDiv.**, **CFreq**, **NClusters**, and **InterDist** corresponds to the methods that use a single feature for ranking. They correspond to context diversity, concept frequency, the number of clusters in the context and inter-cluster distance respectively. During the training, we also conducted the 5 fold cross-validation on PI. The average MAP@10 measure of the proposed method (i.e., combining all feature) is 0.334. It is clear that combining all the features can consistently and significantly outperform the baseline method over all the test collections.

Table 4 shows the testing performance for the three test collections. Note that the parameters are set based on the values learned on the training set, i.e., PI. **All** still denotes the performance of combining all the features, and **BL** denotes the best performance when using a single feature. Moreover, the learned parameters on the training set seem to work well on the other test sets even if they are from completely different domains.

Table 3. Optimal Performance for *Overloaded* Detection on Training Set (PI)

Features	MAP@10	P@10	R@10
All	0.42	0.5	0.63
CDiv	0.24	0.3	0.38
CFreq	0.27	0.4	0.5
NClusters	0.21	0.3	0.38
InterDist	0.19	0.2	0.25

Table 4. Test Performance Comparison for *Overloaded* Detection

	All			BL		
	MAP@10	P@10	R@10	MAP@10	P@10	R@10
PII	0.21	0.6	0.26	0.11	0.5	0.22
PIII	0.15	0.2	0.18	0.08	0.1	0.09
PIV	0.42	0.4	0.57	0.12	0.1	0.14

The similarity boundary is used as the stop criterion of the HAC method, i.e., when the maximum similarity value of two clusters is smaller than the similarity boundary, the clustering procedure stops. Therefore, the value of the similarity boundary affects the performance of *NCluster*, *InterDist* and *All* for overloaded ambiguity detection. We now examine the performance sensitivity with respect to the value of similarity boundary. Figure 2 shows the sensitivity curves for all the three methods on the training collection (i.e., PI). It is clear that the similarity boundary can not be either too large or too small. When the similarity boundary is too large, we may separate similar contexts into different groups. On the other hand, when the similarity boundary is too small, we may not be able to distinguish different contexts. For example, if the threshold is 0.1, most of the contexts will be grouped together and the ability to differentiate them is not limited. Our preliminary results suggest that the optimal value for the similarity boundary is around 0.3.
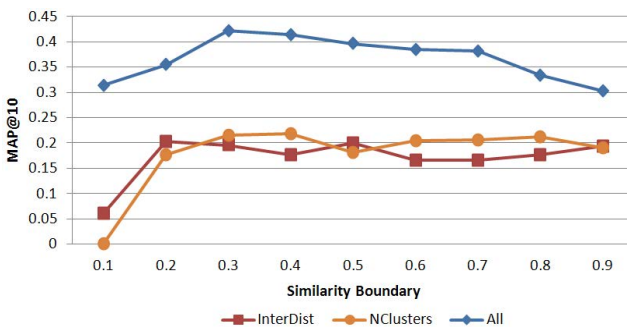
Fig. 2. Similarity boundary affects the performance(Project I)

7.2   Effectiveness of Synonymous Ambiguity Detection

We also evaluate the effectiveness of synonymous ambiguity detection methods. Table 5 shows the optimal performance on the training set **PI**. We conducted the 5-fold cross-validation for synonymous detection too. The average MAP@10 for using all features is 0.3. It is clear that using all the features is more effective than using individual features. In particular, using the textual-based feature outperform using the other two features. Furthermore, it is worth noticing that the context information is useful in detecting overloaded ambiguous concepts (all the features used in overloaded detection is based on context of the concept) but not helpful in detecting synonymous ones. The Contexts-based Similarity method does not perform as well as we expected. The reason, to our understanding, is because of the concept co-occurrence problem. Although currently the penalty is applied on the terms that show together, it is possible that two different concepts show in similar contexts but are not synonymous. On the other hand, it is not surprising to find that textual-based Similarity has a better performance, because similar concepts often share common terms.

Table 5. Optimal Performance for *Synonymous* Detection on Training Set (PI)

Feature	MAP@10	P@10	R@10
All	**0.31**	**0.4**	**0.44**
Textual-based	0.13	0.2	0.22
Context-based	0.07	0.2	0.33
Pattern-based	0.11	0.1	0.11

Table 6. Test Performance Comparison for *Synonymous* Detection

	All			BL		
	MAP@10	P@10	R@10	MAP@10	P@10	R@10
PII	**0.38**	**0.2**	**0.66**	0.16	0.1	0.33
PIII	**0.17**	**0.3**	**0.42**	0.09	0.3	0.42
PIV	**0.37**	**0.3**	**0.5**	0.13	0.2	0.33

With the parameters trained on Project I, we report the test performance on the other three collections in Table 6. **BL** denotes the baseline method using a single feature, and we use the textual based feature in this set of experiments since it is more effective than the other two features. Results show that it is more effective to combine all the features, and the conclusion holds for all the test sets.

We also conduct an exit survey with assessors and ask them about their experience in making the judgments for synonymous ambiguity detection. We find that it takes more efforts to make judgments for this ambiguity type, and it is necessary to consider both context and semantic meaning of the concepts to detect such ambiguities. Furthermore, the assessors also state that the ranked list is a good tool that can help them identify the ambiguous pairs more effective. In particular, the pairs remind them of some concepts that could be interchangeable, which was really helpful, especially when the SRS is long.

7.3   Discussions

Identifying ambiguous concepts from natural language is a difficult task, even for human assessors. To demonstrate that, we evaluated the judgment results from assessors. As every project have 3 sets of judgment, one of them is chosen as the golden standard to evaluate the remaining two. We iteratively conducted this evaluation in the project, and reported the average performance as shown in table 7. It is worth to notice that the performance of the manually created results is only around 0.5 for MAP. This low value proved that ambiguity detection is a challenging tasks even for well trained human assessors.

Table 7. Evaluation of manually created results

	Overloaded			Synonymy		
	MAP@10	**P@10**	**R@10**	**MAP@10**	**P@10**	**R@10**
PI	0.53	0.61	0.79	0.49	0.61	0.61
PII	0.49	0.78	0.49	0.46	0.68	0.76
PIII	0.47	0.48	0.50	0.36	0.61	0.38
PIV	0.52	0.51	0.53	0.57	0.58	0.67

8   Conclusions and Future Work

Our paper is one of the first papers that aim to detect ambiguous terminology from software requirements specifications. The problem is important yet under-studied. To tackle the challenge, we propose to formulate the problem as a ranking problem, and then discuss how to estimate the overloaded ambiguity scores for concepts and synonymous ambiguity scores for concept pairs. Experiment results over four real-world data sets show that the proposed combined methods are more effective than those methods using single features alone, and they have potential to help software engineers to detect ambiguity terminologies more efficiently.

Another interesting outcome from a software engineering perspective is the abundance of ambiguous terminology found in the four SRSs we used in the evaluation. The ambiguities were identified through a manual process, and averaged around 20% of concepts per SRS are ambiguous either because they are overloaded or synonymous. The large number of ambiguous concepts present, really reinforces the need for automated techniques that can help reduce these ambiguities and produce more consistent SRSs.

There are a few interesting directions for the future work. First, we plan to study how to automatically learn the weights for the proposed combined method based on the statistics of the data sets. Second, the detection performance is closely related to the quality of extracted concepts. We will study other concept extraction methods and see whether they are improve the detection performance. Finally, it would be interesting to study other types of ambiguities such as the scope ambiguity and attachment ambiguity [1,2].

References

1. Berry, D.M.: Ambiguity in natural language requirements documents. In: Paech, B., Martell, C. (eds.) Monterey Workshop 2007. LNCS, vol. 5320, pp. 1–7. Springer, Heidelberg (2008)
2. Berry, D.M., Kamsties, E., Krieger, M.M.: From contract drafting to software specification: Linguistic sources of ambiguity (2003),
   http://se.uwaterloo.ca/~dberry/handbook/ambiguityHandbook.pdf
3. Boehm, B.W., Papaccio, P.N.: Understanding and controlling software costs. IEEE Transaction of Software Engineering 14, 1462–1477 (1988)
4. Chantree, F., Nuseibeh, B., de Roeck, A., Willis, A.: Identifying nocuous ambiguities in natural language requirements. In: Proceedings of the 14th IEEE International Requirements Engineering Conference, Washington, DC, USA, pp. 56–65 (2006)
5. Cobleigh, R.L., Avrunin, G.S., Clarke, L.A.: User guidance for creating precise and accessible property specifications. In: ACM SIGSOFT 14th International Symposium on Foundations of Software Engineering, pp. 208–218 (2006)
6. Damas, C., Lambeau, B., Dupont, P., van Lamsweerde, A.: Generating annotated behavior models from end-user scenarios. IEEE Transaction of Software Engineering 31, 1056–1073 (2005)
7. Frantzi, K., Ananiadou, S.: Extracting nested collocations. In: Proceedings of the 16th Conference on Computational Linguistics, vol. 1, pp. 41–46 (1996)
8. Greenspan, S., Mylopoulos, J., Borgida, A.: On formal requirements modeling languages: Rml revisited. In: Proceedings of the 16th International Conference on Software Engineering, Los Alamitos, CA, USA, pp. 135–147 (1994)
9. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th Conference on Computational Linguistics, Stroudsburg, PA, USA, vol. 2, pp. 539–545 (1992)
10. Hussain, I., Ormandjieva, O., Kosseim, L.: Automatic Quality Assessment of SRS Text by Means of a Decision-Tree-Based Text Classifier. In: Seventh International Conference on Quality Software (QSIC), pp. 209–218 (2007)
11. Ide, N., Véronis, J.: Word sense disambiguation: The state of the art. Computational Linguistics 24, 1–40 (1998)
12. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
13. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. MIT Press, Cambridge (1999)
14. Maynard, D., Funk, A., Peters, W.: Using lexico-syntactic ontology design patterns for ontology creation and population. In: Proceedings of WOP 2009 Collocated with ISWC 2009, vol. 516 (2009)
15. Nikora, A., Hayes, J., Holbrook, E.: Experiments in Automated Identification of Ambiguous Natural-Language Requirements. In: Proc. 21st IEEE International Symposium on Software Reliability Engineering, San Jose
16. Porter, A., Votta, L.: Comparing detection methods for software requirements inspections: A replication using professional subjects. Empirical Software Engineering 3, 355–379 (1998)
17. Porter, A.A., Votta Jr., L.G., Basili, V.R.: Comparing detection methods for software requirements inspections: A replicated experiment. IEEE Transaction of Software Engineering 21, 563–575 (1995)
18. Reubenstein, H.B., Waters, R.C.: The requirements apprentice: an initial scenario. SIGSOFT Software Engineering Notes 14, 211–218 (1989)
19. Roark, B., Charniak, E.: Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction. In: Proceedings of the 17th International Conference on Computational Linguistics, Stroudsburg, PA, USA, vol. 2, pp. 1110–1116 (1998)

20. Shull, F., Rus, I., Basili, V.: How perspective-based reading can improve requirements inspections. Computer 33, 73–79 (2000)
21. Tratz, S., Hovy, D.: Disambiguation of preposition sense using linguistically motivated features. In: HLT-NAACL (Student Research Workshop and Doctoral Consortium), pp. 96–100 (2009)
22. Umber, A., Bajwa, I.S.: Minimizing ambiguity in natural language software requirements specification. In: Digital Information Management (ICDIM), pp. 102–107 (2011)
23. van Lamsweerde, A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. John Wiley & Sons (2009)
24. Zhang, X., Fang, A.: An ATE system based on probabilistic relations between terms and syntactic functions. In: 10th International Conference on Statistical Analysis of Textual Data - JADT 2010 (2010)
25. Zou, X., Settimi, R., Cleland-Huang, J.: Improving automated requirements trace retrieval: a study of term-based enhancement methods. In: Empirical Software Engineering, vol. 15, pp. 119–146 (2010)
26. Zowghi, D., Gervasi, V., McRae, A.: Using default reasoning to discover inconsistencies in natural language requirements. In: Proceedings of the Eighth Asia-Pacific on Software Engineering Conference, Washington, DC, USA, pp. 133–140 (2001)

An OpenCCG-Based Approach to Question Generation from Concepts

Markus M. Berg[1,2], Amy Isard[3], and Johanna D. Moore[3]

[1] Department of EE & CS, Wismar University, Germany
mail@mmberg.net
[2] Department of Computer Science, Kiel University, Germany
[3] School of Informatics, University of Edinburgh, Scotland, UK
{amy.isard,j.moore}@ed.ac.uk

Abstract. Dialogue systems are often regarded as being tedious and inflexible. We believe that one reason is rigid and inadaptable system utterances. A good dialogue system should automatically choose a formulation that reflects the user's expectations. However, current dialogue system development environments only allow the definition of questions with unchangeable formulations. In this paper we present a new approach to the generation of system questions by only defining basic concepts. This is the basis for realising adaptive, user-tailored, and human-like system questions in dialogue systems.

1   Introduction

Speech-based information systems offer dialogue based access to information via the phone. They avoid the complexity of computers/websites and introduce the possibility of accessing automated systems without being distracted from your visual focus (e.g. while driving a car) and without needing your hands or eyes (e.g. for visually impaired people or workers with gloves). Most people have already used spoken dialogue systems (SDS) in order to reserve tickets for the cinema, to check the credit of a pay-as-you-go phone or to look for the next bus. Generally, a SDS can be defined as a system that *"enables a human user to access information and services that are available on a computer or over the Internet using spoken language as the medium of interaction"* [11]. In this way, these systems can offer a convenient way of retrieving information. However, Bringert [5] identifies three major problems with current interactive speech applications: They are not natural, not usable and not cheap enough. Berg [2] found that 71% of users prefer the most natural dialogues when choosing from three fictional human-machine dialogues. This is in line with the results of Dautenhahn et al. [7], who also found that 71% of people wish for a human-like communication with robots. Looi and See [14] describe the stereotype of human-robot dialogue as being monotonous and inhumane. They argue that the engagement between human and robot can be improved by implementing politeness maxims, i.e. connecting with humans emotionally through a polite social dialogue. In order to realise user-friendly and natural dialogue systems

that apply politeness maxims and adapt their style to the user's language, we need support from a language generation component.

In this paper we describe a method for generating system questions in information-seeking dialogue systems. Our aim is to formulate these questions in different styles (formality and politeness) from abstract descriptions (*concept-to-speech*). We hope to increase the user acceptance of dialogue systems by contributing to a method for generating human-like and adaptive utterances.

2   Related Work

As this paper focusses on the generation of questions in different styles, we review related work in the areas of question generation and linguistic style.

2.1   Question Generation

Question Generation is a twofold area of research. On the one hand, it deals with text understanding and the generation of questions related to the content. This area is of special interest for tutoring system researchers, where the system has to understand the content of an article, identify relevant sentences and formulate questions about them in order to automatically create reading assessments. On the other hand, question generation can be seen as a subdomain of language generation and *concept-to-speech* technology. Especially in rapid application development, the specification of concepts along with surface parameters instead of hard-coded system prompts reduces development effort and introduces the possibility of adapting system utterances with regard to the demands of the user. Based on the contributions to the *Workshops on Question Generation* in 2009 [21] and 2010 [4], most work has been done in the area of tutorial dialogue, i.e. question generation from text. Papasalouros [19] describes how to generate multiple choice questions for online examination systems, Ou et al. [18] show how to extract predictive questions from an ontology that might be asked by the user of a question-answering system, and Olney et al. [17] describe an approach to generate questions by filling templates. Their approach also focusses on tutorial dialogue and is based on psychological theories that claim that questions are generated from a concept map. But as the use of representations has been avoided in state of the art question answering systems, Olney et al. discuss whether a representation-free approach that bases on syntax transformations (e.g. wh-fronting) of given declarative sentences can successfully generate questions. They found that this approach has difficulties with determining the question type and conclude that a certain degree of knowledge representation is necessary.

2.2   Linguistic Style

Mairesse [15] explains linguistic style as *"a specific point within the space of all possible linguistic variation"* and concludes that it can be considered as a *"temporary characteristic of a single speaker"*. He refers to Brown et al. [6]

and the need for self-esteem and respect from others and states *"that the use of politeness is dependent on the social distance and the difference of power between conversational partners, as well as on the threat of the speakers communicative act towards the hearer"*. Gupta et al. [8] state that *"Politeness is an integral part of human language variation, e.g. consider the difference in the pragmatic effect of realizing the same communicative goal with either 'Get me a glass of water mate!' or 'I wonder if I could possible have some water please?' "*. Jong et al. [12] claim that language alignment happens not only at the syntactic level, but also at the level of linguistic style. They consider linguistic style variations as an important factor to give virtual agents a personality and make them appear more socially intelligent. Also Raskutti and Zukerman [20] found that naturally occurring information-seeking dialogues include social segments like greeting and closing. Consequently, Jong et al. [12] describe an alignment model that adapts to the user's level of politeness and formality. The model has three dimensions: politeness, formality and T-V distinction. Whereas politeness is associated with sentence structures, formality is dependent on the choice of words. In many languages we have to differentiate between a formal and an informal addressing (T-V distinction) of people. This feature is clearly related to both formulation and politeness. However, in Jong's model this feature is not being influenced by formality or politeness changes during the conversation in order to prevent the dialogue from constantly switching between both extrema.

3   Style Variation

Style variation is the generation of different formulations with the same goal. In this paper we focus on task-oriented dialogue systems. This class comprises question-answering-, command-, and information-seeking/booking systems [16]. In particular, we regard the style variation of *interrogatives*. We use this term in order to refer to all kinds of utterances that have the aim of getting information. This may be a question (*"When do you want to leave for London?"*) or a request (*"Tell me when you want to leave for London!"*). Both interrogatives have the same intention, i.e. getting the time of departure.

3.1   Question Types

When trying to classify questions, we have to differentiate between in-tention and style. When classifying by intention, both interrogatives from the last example should be of the same type. Berg et al. [3] describe an Abstract Question Description (AQD) that consists of the answer type, reference type, purpose, surface modifier and cardinality. This would result in `AQD = (fact.temporal.date, fact.namedEntity.nonAnimated.location.city, gather information, positive, 1)`, i.e. an interrogative that expects a date as answer, refers to a city, is meant to get new information from the user, refers to a positive formulation, and expects a single answer. This very abstract definition is useful when it comes to the modelling of dialogue systems. Imagine an integrated development environment that allows you to define

concepts from which the system should generate questions, instead of formulating static and inflexible questions as strings. In this scenario, AQDs can also formalise the parser (in this case a date grammar could be provided). They also help the language understanding component by reducing differently formulated utterances with the same goal to a common description.

For question generation, however, we need more information about the style. While the classification by question words has been declared impractical for describing the intention of an interrogative because different question words can refer to the same goal, e.g. *when* and *at what time* [3], it is still important for style variation. Hence, the relation between AQD and question word can be useful for choosing the correct question word. Apart from the question word, we can also vary grammatical characteristics. When generating a question for the AQD answer type `fact.temporal.date`, we can think of different formulations:

Wh-Question: *When do you want to go?*
Wh-Request: *[Please] Tell me / specify when you want to go!*
NP-Request: *[Please] Tell me your departure time [please]!*
C-Wh-Question: *Can/Could you [please] tell me when you want to go?*
C-NP-Question: *Can/Could you [please] tell me your departure time?*
Command (NP): *Your departure time?*
Command (N): *Departure time...?*

We clearly see that all these utterances have the same intention. However, the AQD is not sufficient for successful generation. In addition to the AQD we also need to define semantic constraints. In this case we could constrain the type of date to departures.

3.2   Politeness and Formality

We have seen different realisations of the same message. But how do these formulations affect us and how is this related to politeness and formality? We have already learned that there exist different degrees of politeness and that systems with an appropriate choice of style are important to the user. However, it is hard to tell what exactly makes an utterance more polite or more formal than another. Moreover, formality and politeness are very close terms that often get mixed. Often a question (*"When do you want to go?"*) seems more polite than a request (*"Tell me when you want to go!"*). But what if we add *please* to the request? Is the request more polite because of this modal particle? Is *can you please* more polite than *could you*? In this paper we work with the following hypothesis, which we plan to evaluate in future user studies: Politeness is characterised by:

- the use of *please*
- the use of a subjunctive modal verb
- an interrogative style (request vs. question)

Formality is sometimes also regarded as influencing politeness, i.e. a very formal style is intended to be polite. In most cases this is true, but nevertheless we

have to distinguish politeness and formality in order to be able to adapt to it independently. We regard formality as the lexicalisation, i.e. the choice of words and word types.

> *Can you tell me when you <u>like</u> to <u>set off</u>?*
> *Can you tell me when you <u>want</u> to <u>leave</u>?*
> *Can you tell me your <u>departure date</u>?*
> *Can you <u>specify</u> when you <u>want</u> to <u>leave</u>?*
> *Can you <u>specify</u> your <u>departure date</u>?*

In this example above we can identify different levels of formality in relation to the choice of words, i.e. *specify* sounds more formal than *tell* and *set off* sounds more colloquial than *leave*. Moreover, the grammatical structure can also influence formality. While the first two sentences use a verb phrase (*you want to leave*), the third one makes use of a noun phrase (*your departure date*). Some languages (e.g. German or French) also differentiate between formal and informal personal pronouns (second person). This is called T-V distinction and affects, as an indicator for social distance, formality. Again, the exact ordering of utterances regarding formality is subject for a user study. We work with the hypothesis that formality can be influenced by:

- the choice of words
- the grammatical form of the sentence (NP vs VP)
- the choice between formal and informal personal second person pronouns

With this distinction between politeness (the use of *please* and subjunctive forms, choice of question style) and formality (choice of words, T-V distinction) we now have parameters at hand to change the style of a system interrogative. In the next section we address the topic of modelling system interrogatives for usage in a toolkit that allows the realisation of interrogatives with respect to given parameters.

4   Realisation

Our aim is the automatic generation of system questions in spoken dialogue systems. We want to provide the development environment with information like *"ask for the departure date in an informal way and don't make special use of politeness"* and it should generate a question like *"When do you want to leave?"*. This process includes several steps and components. As you can see in Figure 1, we first need to define the communicative goal. This goal needs to be connected to the content (what should be said) and the form (how it should be formulated). This is known as lexicalisation (choice of words and grammatical style) and dependent on the parameters for politeness and formality. This decision has to be made in close connection to the content, i.e. we need to find a representation that provides us with information about word meaning, formality and politeness. Moreover, we need a set of rules for the generation of different question types.
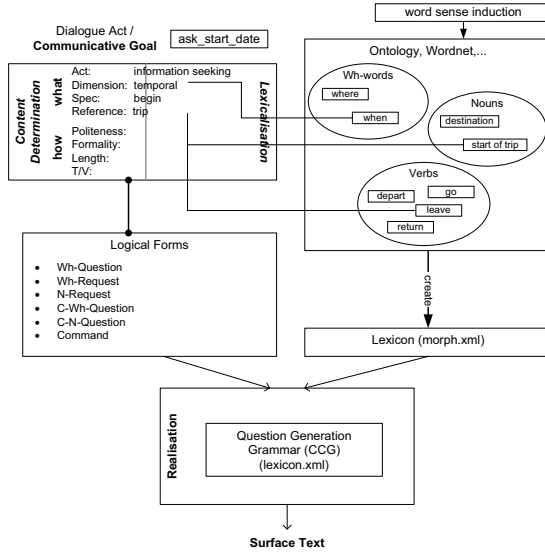
Fig. 1. System overview

With this information we can create logical forms that can be used as input for our language generator. We first take a closer look at the language generation process. Afterwards we describe our knowledge base and how we can use the results in dialogue systems.

4.1 Combinatory Categorial Grammar

Mark Steedman's Combinatory Categorial Grammar (CCG) formalism contains no phrase structure rules and consists only of lexical entries. Steedman and Baldridge [22] describe this as *"a form of lexicalized grammar in which the application of syntactic rules is entirely conditioned on the syntactic type, or category, of their inputs"*. These categories *"may be either atomic elements or (curried) functions which specify the canonical linear direction in which they seek their arguments"* [1]. Atomic elements (sometimes called saturated), such as N, NP, PP or S, do not need to be combined with other elements. They are complete in themselves. With the help of the slash operator we can combine saturated and unsaturated elements to new saturated or unsaturated elements. The slash indicates on which side the arguments should appear. This leads to forward (argument on the right) and backward (argument on the left) application. We call unsaturated elements functions because a complex category like Y/X denotes an element that looks for an X on the right in order to become an Y. This is exactly the behaviour of a function that takes X as an argument and returns Y as result.

$$\frac{Y/X \quad X}{Y}> \qquad\qquad \frac{X \quad Y\backslash X}{Y} < \qquad\qquad (1)$$

Apart from application combinators, there are composition and type-raising combinators. Composition refers to the combination of two functions where the domain of one is the range of the second. It is described with the operator B together with the application direction. Type-raising turns *"arguments into functions over functions-over-such-arguments"* [22], i.e. argument X is turned into a function that has a complex argument that takes this X as an argument. It allows *"arguments to compose with the verbs that seek them"* and is also used in order to be able to apply all rules into one direction (i.e. incremental processing; a full left-to-right proof). Type-raising is denoted with a T. Given the following lexicon, we can derive the sentence *"I like science"* as in (2), or with type-raising and composition as a full left-to-right-proof as in (3).

$$
\begin{array}{lll}
science & \vdash & NP \\
I & \vdash & NP \\
like & \vdash & (S\backslash NP)/NP
\end{array}
$$

$$
\begin{array}{ccc}
i & like & science \\
\mathrm{np} & \mathrm{s\backslash np/np} & \mathrm{np} \\
\hline
 & \mathrm{s\backslash np} & > \\
\hline
\mathrm{s} & & <
\end{array} \quad (2)
$$

$$
\begin{array}{ccc}
i & like & science \\
\mathrm{np} & \mathrm{s\backslash np/np} & \mathrm{np} \\
\hline
\mathrm{s/s\backslash np} & >\mathbf{T} & \\
\hline
 & \mathrm{s/np} & >\mathbf{B} \\
\hline
 & \mathrm{s} & >
\end{array}
$$

$$(3)$$

OpenCCG[1] is a collection of natural language processing tools, which provide parsing and realisation support based on the CCG formalism. A set of XML-based files allows us to define the lexicon and the categories. For a deeper introduction to the OpenCCG syntax you may refer to [13].

4.2   Definition in OpenCCG

In order to generate different formulations, we first have to define a grammar. In this paper we restrict ourselves to the following interrogative types: Yes-No-Question, Wh-Question, Wh-Request, NP-Request, Can-Wh-Question, Can-NP-Question and Command (NP). As already mentioned in the previous section, a categorial grammar does not consist of phrase structure rules that define how a wh-question or a wh-request can be created. Instead, we define a lexicon that describes how every word type can be combined, i.e. how categories can be aggregated to a more general category. A wh-word can be used to create an ordinary question (*"When do you want to go?"*) or can also be part of an indirect request (*"Can you tell me when you want to go?"*):

```
when ⊢
    s[wh-question]/s[question] ▷ question
    s[iwh-question]/s[b] ▷ indirect request
```

---

[1] http://sourceforge.net/projects/openccg/

In our example we apply the first category[2]. This can be read as: *The word 'when' can become a sentence of type 'wh-question' if there is a sentence of type 'question' on the right-hand side.* Since the definition of wh-words is not enough, we now take a look at how the right-hand category of the rule is defined. A question can be created with the help of an auxiliary verb like *do*:

```
do ⊢
    s[question]/s[b]
```

The word 'do' can become a sentence of type 'question' if there is a sentence with a bare infinitive on the right. In order to create such a sentence, we need a verb. Generally, an intransitive verb is defined as `s\np`. The feature `b` denotes a *bare infinitive* and `to` is an *infinitive with to* [10, 9]. The combination of the lexems *want*, *to*, and *go* leads to a category that becomes a sentence with a bare infinitive if there is a `np` on the left. Figure 2 shows the complete application of the CCG categories.

```
(lex)    when :- s[wh-question]/s[question]
(lex)    do :- s[question]/s[b]
(lex)    you :- np
(lex)    want :- (s[b]\np)/(s[to]\np)
(lex)    to :- s[to]\np/(s[b]\np)
(lex)    go :- s[b]\np
(>)      to go :- s[to]\np
(>)      want to go :- s[b]\np
(<)      you want to go :- s[b]
(>)      do you want to go :- s[question]
(>)      when do you want to go :- s[wh-question]
```

Fig. 2. Simplified parse for 'When do you want to go?'

In this example we have described the general methodology to generate wh-questions. However, we don't want to parse sentences with the grammar but instead want to generate (or realise) them. The basis for the OpenCCG realisation process is logical forms, i.e. semantic representations of the sentence. Each syntactic category is associated with a logical form represented in a *hybrid logic dependency structure* that describes the relations between the words of a sentence; e.g. the transitive verb *buy* can be described as:

$$@e\ buy,\ @e{<}sbj{>}\ x,\ @e{<}obj{>}\ y$$

The verb *buy* has two relations, subject and object. The advantage of this semantic description is its independence from the syntactic form. The sentences *"Peter buys a bike"* and *"A bike is bought by Peter"* can be represented as:

---
[2] Brackets indicate features that need to be unified.

$@e(buy \wedge$
$\quad\quad <sbj> (x \wedge Peter)\wedge$
$\quad\quad <obj> (y \wedge bike))$

Peter buys a bike   A bike is bought by Peter
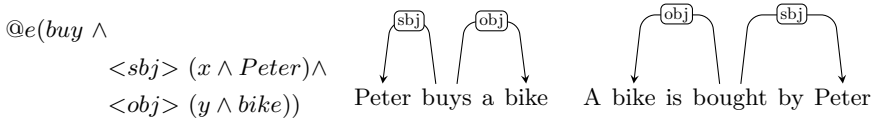
The logical form for the wh-question from our last example is depicted in Figure 3. We use thematic roles in order to specify the proposition of a question. In this case an *agent* wants a *theme*. We can also use semantic features to influence the style of the utterance, i.e. in this case we want to formulate an interrogative sentence with a second person singular agent. As already mentioned, logical forms

```
s{stype=wh-question}:
  @w0(when  ^
      <prop>(w3  ^  want  ^
             <mood>interrogative  ^
             <agent>(w2  ^  pron  ^
                     <num>sg  ^
                     <pers>2nd)  ^
             <theme>(w5  ^  go  ^
                     <agent>x1)))
```

Fig. 3. Logical form for 'When do you want to go?'

are the basis for the realisation process. It abstracts from grammar and word position issues and just reflects the logical meaning of an utterance. Additionally, we need to know which words to use, i.e. OpenCCG requires a finished lexicalisation process. While there are words that are only influenced by inflection (pronoun, sg, 2nd), we also have words that change the style of an utterance[3]. Another way of realising the same meaning with a different lexicalisation would have been *"When do you want to leave?"*.

5   Concept to Text

As already mentioned, our aim is the automatic generation of system questions in an information-seeking dialogue system. We want to be able to instruct the system to create *a question that asks for the departure time in a polite but informal way* without mentioning specific words. This is absolutely necessary to create different levels of formality. So instead of defining words in the logical

---

[3] Indicated with bold face in Figure 3.

form we need meaning representations, i.e. we have to replace the bold faced words in Figure 3 with concepts.

5.1   Word Meaning

In order to describe question words, we can use a slightly modified version of Berg's AQD hierarchy [3], e.g. *"when"* can be described by `fact.temporal`, *"where"* by `fact.named_entity.non-animated.location` and *"at what time"* by `fact.temporal.time`. For verbs we need more information than just the dimension. One approach is the description of the meaning of a word by combining conjunctive features:

– go: movement $\wedge$ slow $\wedge$ by feet
– run: movement $\wedge$ fast $\wedge$ by feet
– drive: movement $\wedge$ by car
– travel: movement $\wedge$ far away $\wedge$ holiday

However, we need a more abstract formulation that also focusses on the similarities of word senses. In a travel domain *go* and *travel* can be synonymous (*"When do you want to go"* = *"When do you want to travel"*) and should therefore have the same description. Thus, as a first draft, we propose to describe a word with its type of usage (or context), so that every word w is assigned a:

– Part of Speech π
– Domain δ
– Context γ: Dimension $\wedge$ Specification
– Referent ρ

which results in $(w \text{ typeof } \pi) \in \delta = (\gamma, \rho)$. In the following examples you can see how we can describe the meaning of words, together with some exemplary sentences. The definition of *go* reads as follows: It is a verb in the travel domain that can be used in a temporal context to describe the beginning of a trip (start date) or in a local context to describe the end of a trip (destination). Nouns follow the same scheme. The only difference is the part of speech.

go (v) $\in$ Travel   **departure city** (n) $\in$ Travel
γ=temporal$\wedge$begin $\vee$ local$\wedge$end γ=local$\wedge$begin
ρ=trip ρ=trip
  ▷ *When do you want to go?*   ▷ *Please tell me your departure city.*
  ▷ *Where do you want to go?*

When we take a look at question words, we can see that we have introduced a *general* domain and a wildcard referent as well as a wildcard specificator.

when (whadv) $\in$ General   **tell** (v) $\in$ General
γ=temporal$\wedge$* γ=*$\wedge$knowledge_transfer
ρ=* ρ=*
   ▷ *When do you want to go?*   ▷ *Can you tell me when you want to go?*

Apart from question words and related verbs or nouns (*when* do you want to *go*) we also have verbs like *want, tell, have* and *can* that cannot be related to an answer type. Here we can use the specificator to denote the context, e.g. a word that can be used in any dimension to indicate a *transfer of knowledge* with reference to any object. With this elementary definition of words we are now able to describe questions. We basically describe a question by γ and ρ, i.e. the context and the referent. The task of creating a question that asks for the begin of a trip would be defined as: $ask(fact.temporal.date, begin, trip)$. According to our question style definitions from section 4.2 and their related grammars, we choose either a verb or a noun to represent γ and ρ. Also neutral words like *want* or *tell* are introduced in this step. In a very formal utterance *tell* could be replaced by *specify*. Apart from the formality, we also have to choose the correct question style according to the politeness. A high politeness value leads to the introduction of the word *please* and changes the mood from indicative to subjunctive. Moreover, we have a list that assigns a politeness value to every question type and thus influences the construction of the logical forms. For example, a can-question is more polite than a request. These values are currently based on intuitions which were backed up by the choices of the evaluation participants (see Section 6) but in future versions we plan to base them on user studies.

The result of this step is the representation in Figure 3, which is – at the same time – the input for the OpenCCG realiser.

5.2   Programming Interface and Results

As mentioned in the beginning, our aim is an easy integration of our approach in current dialogue systems. The programmer should not have to deal with complicated language generation issues. Instead, he should be able to use a simple interface. The management of the vocabulary is handled by an ontology. The following lines of code generate a dialogue with five system questions, that ask for the start date of the trip, the end date, the departure city, the destination and whether the customer has a customer card:

When setting the formality value to 2 and the politeness value to 1, we achieve the result shown in Dialogue 1. As you can see, every second utterance we insert a temporal connector (*now*) to make the dialogue appear more fluent. In Dialogue

```
1 int intended_formality=2;  //1..5
2 int intended_politeness=1;  //-2..5
3 questions.add(new Question("fact.temporal.date", "begin", "trip"));
4 questions.add(new Question("fact.temporal.date", "end", "trip"));
5 questions.add(new Question("fact.location", "begin", "trip"));
6 questions.add(new Question("fact.location", "end", "trip"));
7 questions.add(new Question("decision", "possession", "customer_card"));
```

2 we have increased the politeness value to 4. You can see that the system now chooses C-Questions and makes use of verbs instead of nouns.

Dialogue 1: *f=2, p=1*

S: Please tell me your departure date!
U: ...
S: Now please tell me your return date!
U: ...
S: Please tell me your departure city!
U: ...
S: Now please tell me your destination!
U: ...
S: Do you have a customer card?
U: ...

Dialogue 2: *f=2, p=4*

S: Can you please tell me when you want to go?
S: Can you now please tell me when you want to return?
S: Can you please tell me where you want to start from?
S: Can you now please tell me where you want to go?
S: Do you have a customer card?

When also increasing the formality to 4, we observe a different choice of words, for example the first utterance may be realised as *"Can you please tell me when you want to depart?"*. We can see that due to the static relationship between formality and question style, almost every utterance has the same formulation within each dialogue. That's why we introduce a *politeness* variation that automatically varies the politeness around a given value, as you can see in Dialogue 3.

Dialogue 3: *f=2, p=1±1*

S: Please tell me your departure date!
S: And when do you want to return?
S: Departure city please!
S: Now please tell me your destination!
S: Do you have a customer card?

6   Evaluation of Generated Example Interrogatives

After having demonstrated the functionality of the proposed system, we now evaluate its plausibility and effects. We have asked 26 human judges to evaluate the dialogues' naturalness and politeness. During the test we presented the participants four dialogues (see Figure 4) with different politeness levels. The users had to sort the dialogues according to their politeness. Afterwards, they had to indicate which of the dialogues might have been uttered by a human and to state which dialogue they prefer.

A	B	C	D
A: When do you want to set off? B: ... A: Can you now tell me when you want to return? B: ... A: Please tell me your departure city! B: ... A: And where do you want to go? B: ... A: Do you have a customer card? B: ...	A: Can you please tell me when you want to set off? B: ... A: Could you now please tell me when you want to return? B: ... A: Can you tell me where you want to start from? B: ... A: Can you now please tell me where you want to go? B: ... A: Do you have a customer card? B: ...	A: Departure date? B: ... A: And the return date? B: ... A: Departure city? B: ... A: And the destination? B: ... A: Customer card? B: ...	A: Departure date please! B: ... A: Now please tell me your return date! B: ... A: Tell me your departure city! B: ... A: And the destination please! B: ... A: Do you have a customer card? B: ...

Fig. 4. Survey

We began with the sorting task. In general the participants correctly classified the dialogues according to our intended politeness levels, and 46% put the dialogues in exactly the right order (C, D, A, B). The participants classified the two more impolite dialogues as more polite than they are, and the two more polite ones as less polite, as shown in Table 1, but this can possibly be explained by people's tendency to choose values in the middle of a scale rather than at either extreme. Now we asked the participants which dialogue they like most. 77% of them preferred dialogue A, 19% preferred dialogue B and 4% dialogue D. When we quantify the preferred dialogues with the corresponding politeness scores and normalise the result (see Equation 4), we get an average preferred politeness score of 3.2 (original score) respectively 2.8 (user score), which again refers to dialogue A.

$$\frac{1}{|user|} \sum_{i=1}^{|dialogues|} score_i \times |votes\_dialogue_i| \tag{4}$$

Table 1. Dialogues sorted by politeness scores

dialogue	original scores	mean user scores	Δ	correctly classified by
C	1.0	1.8	+0.8	62%
D	2.0	2.2	+0.2	62%
A	3.0	2.7	-0.3	58%
B	4.0	3.3	-0.7	65%

In the last step we asked the users to indicate which dialogue might have been uttered by a human. 88% of the participants think that dialogue A could have been uttered by a human, 42% think dialogue B might be of human origin, 19% declare dialogue C and 4% dialogue D as human. This evaluation confirms that the system is able to create questions in different politeness levels and that these levels are correctly identified by the users.

7   Conclusion and Future Work

We have proposed a model to create system utterances in different politeness and formality levels, which serves as our basis for improving dialogue systems and dialogue development environments with respect to adaptive and human-like formulations of system utterances. Based on a categorial grammar we have created seven different interrogative utterance types that represent the syntactic form for different politeness values. An ontology defines the meanings and formality levels of the used words. To verify the operationality of our model, we have developed a programming interface that is used to define concepts which are then realised according to the politeness and formality parameters. A final evaluation of the generated utterances proves that our model is a valid methodology to support concept-to-text with respect to the generation of system questions.

In our future work we want to extend this model to more complicated utterances and also to support the semi-automatic parser generation from the existing concepts and corresponding answer types. Another area of research is, apart from the realisation of questions, the generation of system answers.

References

1. Baldridge, J., Kruijff, G.J.M.: Multi-modal combinatory categorial grammar. In: Proceedings of the Tenth Conference on EACL, Stroudsburg, PA, USA, pp. 211–218 (2003)
2. Berg, M.M.: Survey on Spoken Dialogue Systems: User Expectations Regarding Style and Usability. In: XIV International PhD Workshop, Wisla, Poland (October 2012)
3. Berg, M.M., Düsterhöft, A., Thalheim, B.: Towards interrogative types in task-oriented dialogue systems. In: Bouma, G., Ittoo, A., Métais, E., Wortmann, H. (eds.) NLDB 2012. LNCS, vol. 7337, pp. 302–307. Springer, Heidelberg (2012)
4. Boyer, K.E., Piwek, P. (eds.): Proceedings of QG 2010: The Third Workshop on Question Generation, Pittsburgh (2010)
5. Bringert, B.: Programming language techniques for natural language applications (2008)
6. Brown, P., Levinson, S.C., Gumperz, J.J.: Politeness: Some Universals in Language Usage. Studies in Interactional Sociolinguistics. Cambridge University Press (1987)
7. Dautenhahn, K., Woods, S., Kaouri, C., Walters, M.L., Koay, K.L., Werry, I.: What is a robot companion - friend, assistant or butler?, pp. 1488–1493 (2005)
8. Gupta, S., Walker, M.A., Romano, D.M.: Generating politeness in task based interaction: An evaluation of the effect of linguistic form and culture. In: Proceedings of the Eleventh European Workshop on NLG, Stroudsburg, PA, USA, pp. 57–64 (2007)

 9. Hockenmaier, J., Steedman, M.: CCGbank: User's Manual. Tech. rep (May 2005)
10. Hockenmaier, J., Steedman, M.: CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. Comput. Linguist. 33(3), 355–396 (2007)
11. Jokinen, K., McTear, M.F.: Spoken Dialogue Systems. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers (2009)
12. de Jong, M., Theune, M., Hofs, D.: Politeness and alignment in dialogues with a virtual guide. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, Richland, SC, pp. 207–214 (2008)
13. Kruijff, G.J.M., White, M.: Specifying Grammars for OpenCCG: A Rough Guide (2005)
14. Looi, Q.E., See, S.L.: Applying politeness maxims in social robotics polite dialogue. In: Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction, pp. 189–190. ACM, New York (2012)
15. Mairesse, F.: Learning to Adapt in Dialogue Systems: Data-driven Models for Personality Recognition and Generation. Ph.D. thesis, University of Sheffield (February 2008)
16. McTear, M.F., Raman, T.V.: Spoken Dialogue Technology: Towards the Conversational User Interface. Springer (2004)
17. Olney, A.M., Graesser, A.C., Person, N.K.: Question generation from concept maps. Dialogue and Discourse 3(2), 75–99 (2012)
18. Ou, S., Orasan, C., Mekhaldi, D., Hasler, L.: Automatic Question Pattern Generation for Ontology-based Question Answering. In: The Florida AI Research Society Conference, pp. 183–188 (2008)
19. Papasalouros, A.: Automatic generation of multiple-choice questions from domain ontologies. Engineer (Bateman 1997) (2008)
20. Raskutti, B., Zukerman, I.: Generating queries and replies during information-seeking interactions. Int. J. Hum.-Comput. Stud. 47(6), 689–734 (1997)
21. Rus, V., Lester, J. (eds.): AIED 2009 Workshops Proceedings: The 2nd Workshop on Question Generation, Brighton (2009)
22. Steedman, M., Baldridge, J.: Non-Transformational Syntax. In: Combinatory Categorial Grammar, pp. 181–224. Wiley-Blackwell (2011)

A Hybrid Approach for Arabic Diacritization

Ahmed Said, Mohamed El-Sharqwi, Achraf Chalabi, and Eslam Kamal

Microsoft Advanced Technology Lab, Cairo, Egypt
{v-ahsaid,moels,achalabi,eskam}@microsoft.com

Abstract. The orthography of Modern standard Arabic (MSA) includes a set of special marks called diacritics that carry the intended pronunciation of words. Arabic text is usually written without diacritics which leads to major linguistic ambiguities in most of the cases since Arabic words have different meaning depending on how they are diactritized. This paper introduces a hybrid diacritization system combining both rule-based and data- driven techniques targeting standard Arabic text. Our system relies on automatic correction, morphological analysis, part of speech tagging and out of vocabulary diacritization components. The system shows improved results over the best reported systems in terms of full-form diacritization, and comparable results on the level of morphological diacritization. We report these results by evaluating our system using the same training and evaluation sets used by the systems we compare against.. Our system shows a word error rate (WER) of 4.4% on the morphological diacritization, ignoring the last letter diacritics, and 11.4% on the full-form diacritization including case ending diacritics. This means an absolute 1.1% reduction on the word error rate (WER) over the best reported system.

Keywords: Arabic, Arabic orthography, diacritization, vowelization, morphology, morphology features, morphological analysis, part-of-speech tagging, automatic correction, Viterbi, case ending, natural language processing, language modeling, conditional random fields, CRF.

1   Introduction

Arabic text in almost all genres of Modern standard Arabic (MSA) is written without short vowels, called **diacritics**. The restoration of these diacritics is valuable for natural language processing applications such as full-text search and text to speech. Devising a diacritization system for Arabic is a sophisticated task as Arabic language is highly-inflectional and derivational. Moreover, Arabic sentences are characterized with a relatively free word-order. The size of the Arabic vocabulary and the complex Arabic morphological structure can both be managed efficiently via working on the morpheme level (constituents of the words) instead of the word level. The system we have built relies heavily on two core components: the morphological analyzer and the part of speech (POS) tagger. By leveraging the systematic and compact nature of Arabic morphology, we have developed a high quality rule-based morphological analyzer with high recall, driven by a comprehensive lexicon and handcrafted rules. Moreover, we have developed a lightweight statistical morphological analyzer that is

trained on LDC's Arabic Treebank corpus (ATB) [8]. The POS tagger is used to re-solve most of the morphological and syntactic ambiguities in context. In the next sections, we present our system as follows: Section 2 covers the linguistic description of Arabic diacritization. Section 3 briefly covers previous related work, in section 4, we elaborate on the different components of the diacritizer, and in section 5, we report our system's results compared to others, using the same evaluation setup: metrics and data.

2   Arabic Diacritization: Linguistic Description

The Arabic alphabet consists of 28 letters; 25 consonants such as "ب" (pronounced /b/) and 3 long vowel letters namelyا , ی (pronounced /a:/), و (pronounced /u:/), and ي (pronounced /i:/). In addition to these letters, there are Arabic diacritics that are classified into 3 groups as shown in Table 1. In order to pronounce any consonant, it has to be associated with one or more of these diacritics.

The first group consists of the short vowel diacritics: *Fatha, Kasra* and *Damma*. Examples of short vowels association with the letter ب are: (i) بَ (pronounced as /b//a/), (ii) بِ (pronounced as /b//i/) and (iii) بُ (pronounced as /b//u/). The second group represents the doubled case ending diacritics (Nunation). These are vowels occurring at the end of nominal words (nouns, adjectives and adverbs) indicating nominal indefiniteness. This phenomenon is called "*Tanween*" and has the phonetic effect of adding an "N" sound after the short vowel at the word ending. *Tanween* applies to the 3 short vowels as follows: (i) *Tanween Fatha* as بًا (pronounced /b//an/), (ii) *Tanween Kasra* such as بٍ (pronounced /b//in/) and *Tanween Damma* as بٌ(pronounced /b//un/). The third group is composed of *Shadda* and *Sukuun* diacrit-ics. *Shadda* reflects the doubling of a consonant, as in بّ (pronounced /b//b/), and is usually combined with a vowel diacritic as in بَّ (pronounced as /b//b//a/). *Sukuun* indicates the absence of a vowel as in بْ (pronounced /b/), and reflects a glottal stop. Diacritics could also be classified into two main categories based on their function. The first category consists of the lexeme diacritics, which determine the part-of-speech of a word as in (ذَ هَبَ → "went" , ذَ هَب → "gold"), and also the meaning of the word such as (رَجُل → "man", رِجْل → "leg"), while the second category re-flects the syntactic function of the word in the sentence, also called case ending. For example, in the sentence "عَرَفَ   اَلـرَّجُلُ   اَلْـحَقِـيقَـةَ", the diacritic "*Fa-tha*" of the word "اَلْـحَقِـيقَـة" reflects its "object" role in the sentence.

While in sentence "وَضَحَتْ   اَلْـحَقِـيقَـةُ" the same word occurs as a "subject" hence its syntactic diacritic is a "*Damma*".

Table 1. Arabic diacritics represented in 3 groups

Diacritic	Diacritic Name	Association with a consonant	Pronunciation
Short vowels (Group 1)			
´	*Fatha*	بَ	/b//a/
	Kasra	بِ	/b//i/
ُ	*Damma*	بُ	/b//u/
Double case ending (tanween)			
ً	*Tanween Fatha*	بً	/b//an/
	Tanween Kasra	بٍ	/b//in/
ٌ	*Tanween Dam-ma*	بٌ	/b//un/
Syllabification marks			
ّ	*Shadda*	بّ	/b//b/
°	*Sukuun*	بْ	/b/

In almost all genres of the Modern standard Arabic (MSA) written text, diacritics are omitted, leading to a combinatorial explosion of ambiguities, since the same Arabic word can have different part-of-speeches and meanings, based on the associated diacritics, e.g.: (عَقْـد → "contract", عِقْـد → "necklace", عَقَّـدَ → "complicate"). The absence of diacritics adds layers of confusion for novice readers and for automatic computation. For instance, the absence of diacritics becomes a serious obstacle to many of the applications including text to speech (TTS), intent detection, and automatic understanding in general. Therefore, automatic diacritization is an essential component for automatic processing of Arabic text.

3   Related Work

We reviewed four approaches in the recent published literature on the diacritization problem; these four publications are the most relevant to our work.

Rashwan et al. [1] designed a stochastic Arabic diacritizer based on a hybrid of factorized and un-factorized textual features. They introduced dual-mode stochastic system to automatically diacritize the raw Arabic text. The first of these modes determines the most likely diacritics by choosing the sequence of full-form Arabic word diacritizations with maximum marginal probability via A* lattice search and long-horizon n-grams probability estimation multilayer Arabic text diacritizer. When full-form words are Out of Vocabulary (OOV), the system resorts to a second mode that factorizes each Arabic word into all its possible morphological constituents, while using the same techniques of the first mode to get the most likely sequence of morphemes, hence the most likely diacritization. While Rashwan et al. [1] is the approach

closest to our work, we introduce new techniques to handle OOVs and generate case endings that leading better results.

Habash and Rambow [2] use a morphological analyzer and a disambiguation system called MADA [4]. They use a feature set including case, mood, and nunation, and use SVMTool [7] as a machine learning tool. They use SRILM toolkit [9] to build an open-vocabulary statistical language model (SLM) with Kneser–Ney smoothing. Habash and Rambow [2] did experiments using the full-form words and the lexemes (prefix, stem, and suffix) citation form. The best results that have been reported are the ones they obtain with the lexemes form with trigram SLM [2]. The system does not handle OOV words which are not analyzed or haven't been seen during training.

Zitouni et al. [3] have built a diacritization framework that based on maximum entropy classification. The classifier is used to restore the missing diacritics on each word letters. They also use a tokenizer (segmenter) and a POS tagger. They use different signals such as the segment n-grams, segment position of the character, the POS of the current segment, and lexical features, including character and word n-grams. Although they don't have a morphological lexicon, they resort to statistical Arabic morphological analysis to segment Arabic words into morphemes (segments). These morphemes consist mainly of prefixes, stems, and suffixes. The maximum entropy model combines all these features together to restore the missing vowels of the input word sequence.

Emam and Fisher [6] introduced a hierarchical approach for diacritization. The approach starts with searching in a set of dictionaries of sentences, phrases and words using a top down strategy. First they search in a dictionary of sentences, if there is a matching sentence, they use the whole text. Otherwise the search starts with another dictionary of phrases, then dictionary of words to restore the missing diacritics. If there is no match at all previous layers, a character n-gram model is used to diacritize each word. No experimental results of this patented work have been mentioned in the available patent document.

The first three systems are trained and tested using LDC's Arabic Treebank (#LDC2004T11) of diacritized news stories text-part 3, v1.0 [8] that includes 600 documents (340 K words) from the Lebanese newspaper "AnNahar". The text is split into a training set (288 K words) and a test set (52 K words) [1], [2], [3]. To our knowledge, these three systems are currently the best performing systems. We adopt their metrics and use the same training and test set for fair comparison.

4   The Hybrid Diacritization System

Our diacritization system is designed as a pipeline of multiple components, each of which addressing a specific aspect of the vowel restoration problem. Figure 1 shows the system's overall architecture, where the diacritization is achieved through 3 main phases: (i) preprocessing, (ii) generation of valid morphological structures (analyses) for each word, combining analyses of both statistical and rule-based morphological analyzers to build a lattice of analyses, and (iii) disambiguation, to generate the diacritized text including case ending diacritics. The preprocessing phase does auto-correction of the raw input text, targeting common Arabic mistakes (CAMs) [5].

Afterwards, the corrected text is tokenized. Each input word is then analyzed through both the statistical and rule-based morphological analyzers, combining zero or more morphological analyses. Each analysis is composed of zero or more prefix(es), the stem, zero or more suffix(es), the morphological pattern, the part of speech tag, and the word tag probability.
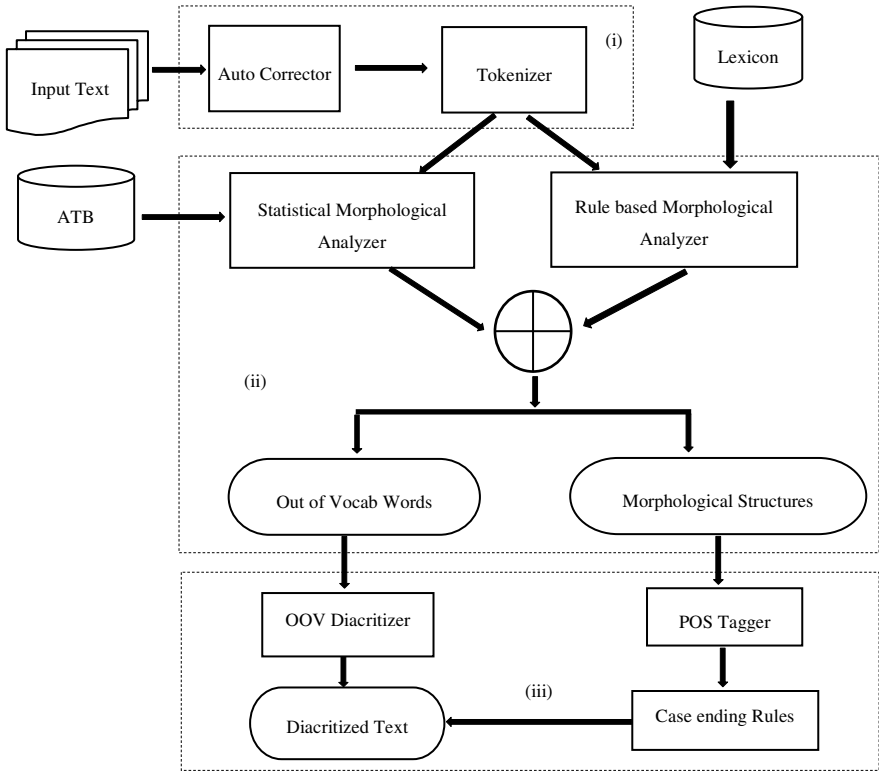
Fig. 1. Architecture of the Arabic Diacritizer

The next phase is responsible for selecting the most likely sequence of analyses based on the context. This is achieved by the POS tagger, which is presented with a lattice of morphological analyses. In addition to selecting the most probable analysis, this process also disambiguates the residual case ending ambiguities.

The case ending diacritics are resolved in two passes: the first pass is a deterministic one and is driven by rules, and the second pass resolves the residual case ending ambiguities through the POS tagger. Out of Vocabulary (OOV) words, those not analyzed by the morphological analyzers, are diacritized by the out of vocabulary diacritizer (OOV diacritizer) component that works on the character level. The contribution of each component has been measured and the outcomes are reported later on the "Results" section.

4.1   Auto-corrector

We conducted a thorough analysis of spelling mistakes in Arabic text. A corpus of one thousand articles, picked randomly from public Arabic news sites, has been semi-manually tagged for spelling mistakes. Each spelling mistake has also an associated error type. The analysis has shown a WER of 6% which is considered very high. The diacritization task is significantly affected by this high error rate since the morphological analyzer fails to analyze misspelt words, and hence are left un-diacritized.

Figure 2 shows the distribution of the spelling mistakes in Arabic text according to our analysis. It was found that more than 95% of these mistakes are classified as CAMs [5], and they could be categorized as follows: (i) Confusion between different forms of *Hamza* (آ إ أ) (ii) Missing *Hamza* on plain *Alef* (ا) (iii) Confusion between *Yaa* (ي) and *Alef-Maqsoura* (ى) , and (iv)Confusion between *Haa* (ه) and *Taa-Marbouta* (ة)
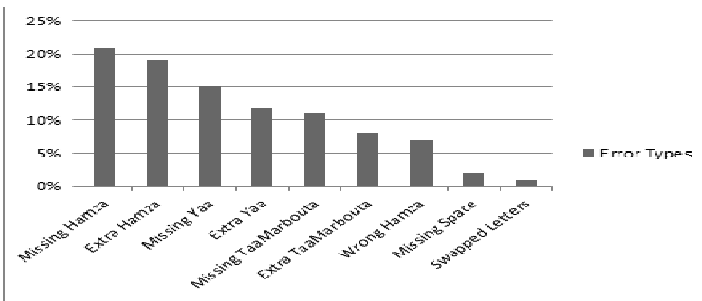
Fig. 2. Distribution of spelling mistakes in Arabic

Relying on a high-precision and high-recall rule-based Arabic morphological analyzer, the Auto-corrector is able to detect and correct most of the common Arabic mistakes automatically with a precision of 99.53% and a recall of 89.32% as illustrated in figure 3
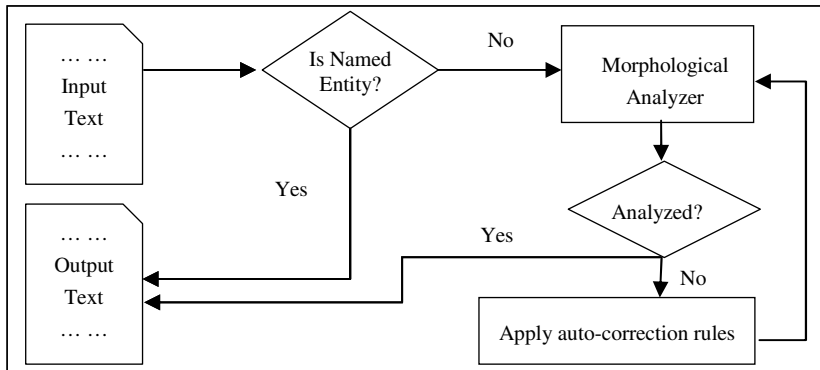
Fig. 3. Autocorrector workflow

4.2   Statistical Morphological Analyzer

We used ATB corpus [8] to train a statistical morphological analyzer that learns the different possible diacritics, the part of speech tags, and the word tag probability:

$$P(w|t) = count(w,t)/ count(t)$$

Where $P(w|t)$ is the word tag probability.

A major drawback of this model is that it suffers a low coverage since the training size is small, considering a highly-inflectional language such as Arabic. Therefore, we combined its output with the rule-based morphological analyzer output to increase the lexical coverage and resolve the problem of unseen words.

4.3   Rule Based Morphological Analyzer

Any valid Arabic word is represented by a morphological structure that consists of zero or more prefixes, a stem, and zero or more suffixes. Prefixes could be conjunctions, prepositions, determiners, and subject pronouns. Suffixes could be subject pronouns, object pronouns, and possessive pronouns. The Arabic stem is the main part of the word after removing any attached prefixes and suffixes. The stem is usually defined by the tuple (r,m,p) where r is the root (usually three or four characters), m is the morphological pattern, and p is the part of speech. The morphological analyzer has been devised based on a **strip-extract-synthesize** approach driven by a comprehensive lexicon, and handcrafted rules. The rules set consists of (956 rules) to extract lexical stems and restore omitted vowels. The lexicon contains: 46,257 stems, 109 part-of-speeches, 3,472 roots, 691 morphological patterns, and 17,186 prefix-suffix combinations.

Morphological analysis of words is achieved in 3 steps as shown in figure 4. The first phase is the stripping phase, where all possible segmentations of the input word (prefixes-raw stem-suffixes) are assumed as valid hypotheses based on a comprehensive prefix-suffix matrix. Then comes the extraction phase where each raw stem (surface string) is subject to extraction rules, to extract the lexical stem. As an example showing the effect of the extraction rules, the word "سمـا و ا ت" smAwAt (using Buckwalter transliteration) where the raw stem " سمـا و" smAw ending with "و" w could be transformed to a lexical stem (real stem) ending with " ء " ‘, provided that the suffix is an "ا ت" At. In the 1st two phases both the applicability between the prefixes and the suffixes and between the affixes and the lexical stems, are checked to eliminate invalid hypotheses. For example, assuming the input word is "بــرق" brq, one possible decomposition would be to consider "ب" b as a preposition prefix and "رق" rq as a past verb stem. However since "ب" b is morphologically incompatible with "رق" rq, then this assumption will be eliminated. The last phase is the synthesis phase, where each remaining hypothesis is subject to synthesis rules, to validate the

hypothesis against the input word. For each valid hypothesis, diacritization rules are applied to restore omitted vowels. For example, if the input word was " مـد رسة أ" mdrspAF ending with *Tanween Fatha* the 1st two phases would propose an analysis where the stem equals " مـد رسة" mdrsp and the suffix equals " أ " AF, however the synthesis phase would reject this assumption since the synthesis rules would actually generate the word " مـد رستـاً" mdrstAF converting " ة " p to "ت" t.
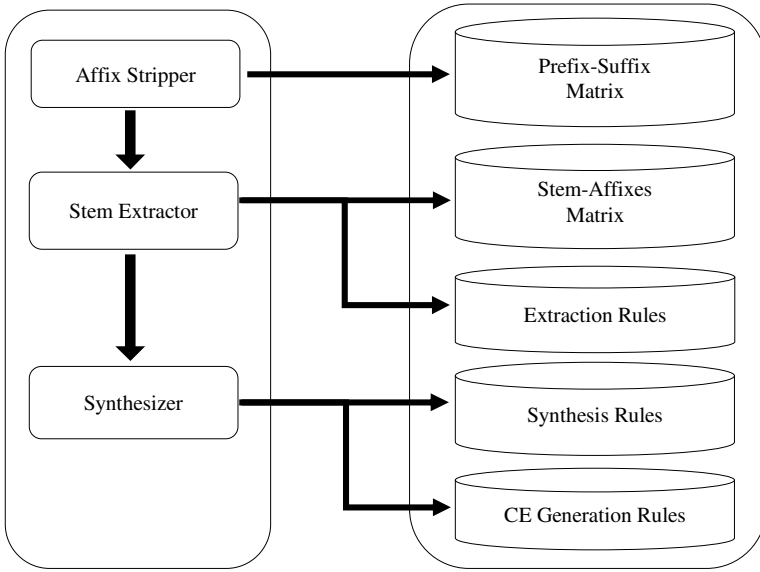
Fig. 4. Morphological Analyzer Architecture

The synthesizer is also responsible for generating the correct case ending diacritic and to position it on the appropriate letter.

Moreover, the morphological analyzer assigns for each generated morphological structure a set of morphological, lexical and syntactic features. Examples of lexical features are transitivity and verb class, morphological features include definiteness, gender and number and examples of syntactic features are case ending, and genitivity.

4.4   Part of Speech Tagger

Part of speech (POS) tagging is the process of assigning a part-of-speech and optionally other syntactic class markers to each word in a sentence. In the context of diacritization, it is used to resolve the morphological ambiguity for each input word in the Arabic sentence, and by resolving this ambiguity, the associated diacritics, on the stem level get automatically resolved. While the morphological analyzer (described in sections 4.2, 4.3) provides all possible valid analyses for each input word, the POS tagger selects the most likely analysis (diacritics) for each word based on the context,

using sequence labeling techniques. We have trained our POS tagger with the ATB corpus [8], the same training set used by Rashwan et al. [1], Habash and Rambow [2], and Zitouni et al. [3].

While in most of the cases, each analysis is mapped to a different tag, sometimes more than one analysis map to the same tag, in such cases, we pick the first analysis.

Table 2. Example of POS Tagging

Index	Word	Translation	POS Tag	Diacritics
1	قـدمت	Presented	PV+PVSUFF_SUBJ:3FS	قَـدَّمَتْ
2	و ر شة	Workshop	NOUN+NSUFF_FEM_SG+CASE_ DEF_NOM	وَ رْشَـةُ
3	عمل	Work	NOUN+CASE_DEF_GEN	عَمَـل
4	الـكتـاب	Book	DET+NOUN+CASE_DEF_GEN	اَلْـكِـتَـاب
5	الـرقـمـي	Digital	DET+ADJ+CASE_DEF_GEN	اَلـرَّقْـمِـيِّ
6	لحـة	Insight	NOUN+NSUFF_FEM_SG+CASE_ INDEF_ACC	لَـمْحَةً
7	عـا مـة	General	ADJ+NSUFF_FEM_SG+CASE_IN DEF_ACC	عَـا مَّـةً
8	عنـه	About it	PREP+PRON_3MS	عَنْـهُ

In addition to the morphological disambiguation, which restores the diacritics on the stem level, the POS tagger also resolves the syntactic ambiguity leading to restoring the syntactic diacritic (case-ending). Table 2 shows the result of POS tagging a short Arabic sentence. The "Index" column in the table is the position of each word in the sentence. The "Word" column is the input word string, the "POS Tag" column is the selected tag by the POS tagger, and each token would initially be presented to the tagger with one or more tag.

The "diacritics" column shows the diacritized word associated with the selected tag, after full disambiguation.

The Hidden Markov Model (HMM) algorithm [11] yields the following function by which the tagger estimates the most probable tag sequence. We used the Viterbi algorithm [11] to compute it. The function contains two kinds of probabilities: word likelihoods and transition probabilities.

$$\hat{t}_1^n \approx \underset{t_1^n}{\mathrm{argmax}} \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1})$$

The POS tagger has been tested on the ATB test set, and it yields accuracy numbers of 86.8% on tag level (including case-ending).

4.5   Out of Vocabulary Diacritizer

OOV words, those for which no single analysis was produced, are classified into 3 main categories: misspelt words, borrowed words (such as "computer" → "كـمـبـيـوتـر") and foreign named entities ("John" → "جون"). For these words, the previously described pipeline is still unable to provide any diacritization alternative. In order to solve this class of words, we developed a statistical model to propose diacritics for foreign named entities, and applied this solution to all OOV words. We have built a training set composed of one thousand foreign named entities, which were manually diacritized, and used this set to train a Conditional Random Field (CRF) model [10]. The features used to train the CRF are shape features, and the resulting accuracy using a 10-fold cross validation is 75.7%. We found that using the OOV diacritizer improves the overall diacritization accuracy by 0.9%.

4.6   Case Ending Rules

The case ending rules comprise 30 rules. These are regular expressions, such as "if current word is a non-ambiguous preposition and the next word is noun, then the second word is genitive". Applying these simple rules has contributed in a 0.7 reduction in terms of WER. These rules are manually revised and tuned over several development sets collected from news sources.

The following is an example of a case ending rule that sets the indicative case ending value on present tense verbs:

IF (WORD @POS 0 CONTAIN STEM "بَيْنَمَا" | "عِنْدَمَا") AND (WORD @POS 1 CONTAIN PREFIX #أحرف_أنيت)
STAMP @POS 1 FLAG "مرفوع"

The rule consists of a **CONDITION** part that holds the conditions and an **ACTION** part that carries the actions to be applied whenever the conditions are satisfied.

5   Results

We adopt the same metrics used by Rashwan et al. [1], Habash and Rambow [2], and Zitouni et al. [3]; also we use the same test set they used to compare our results with the three systems. The metrics that were used are:

1. Count all words, including numbers and punctuation.
2. Each letter or digit in a word is a potential host for a set of diacritics.
3. Count all diacritics on a single letter as a single binary choice.
4. Non-variant diacritization (stem level) is approximated by removing all diacritics from the final letter (Ignore Last), while counting that letter in the evaluation.

Two error rates are calculated: the diacritic error rate (DER), which represents the number of letters which diacritics were incorrectly restored, and the WER representing the number of words having at least one DER.

<p align="center">Table 3. Diacritization results, comparing to the best performing systems</p>

Model	Full form		Ignore last	
	WER	DER	WER	DER
Rashwan et al.	12.5%	3.8%	3.1%	1.2%
Habash et al.	14.9%	4.8%	5.5%	2.2%
Zitouni et al.	18.0%	5.5%	7.9%	2.5%
Our System (All Layers)	11.4%	3.6%	4.4%	1.6%
Our System (Disable POST)	40.5%	10.7%	6.5%	2.3%
Our System (Disable Rules)	12.1%	3.8%	No effect	No effect
Our System (Disable OOV diacritizer)	12.3%	4 %	5.2%	2 %

As depicted in Table 3, our system provides the best results in terms of WER on the full form level, and shows comparable results on the other metrics. We found that the test set has a noticeable number of errors such as (misspelt words, colloquial words, undiacritized words, wrong diacritization)[1]. We also tested our system against another blind test set (TestSet2) consisting of 1K sentences that we collected from different sources and had them manually diacritized. Out of this test set, we derived two other test sets: one for full-form diacritization and another one for morphological diacritization. It's worth mentioning here that the way the "ignore last" metric is handled is not always linguistically correct. In many cases, the syntactic diacritics do not show on the last letter and rather appear on the last letter of the stem as in " مَـدْ رَسَـتَـهُ ". The actual case ending here is the *Fatha* appearing on the before-last letter "ت". In Table 4 below, the first two rows show the results of our system using both the ATB test set, and TestSet2. The remaining rows in the table show the results of our system on TestSet2 after disabling the POS tagger, the case ending rules, the NED, and the auto corrector respectively.

<p align="center">Table 4. Blind test set results</p>

System	Test set	Full form		Morphological Diacritization	
		WER	DER	WER	DER
All Layers enabled	ATB	11.4%	3.6%	4.4%	1.6%
All Layers enabled	TestSet2	8%	2%	2.5%	0.8%
Disable POST	TestSet2	52%	11%	3.2%	1.1%
Disable CE Rules	TestSet2	8.6%	2.3%	No effect	No effect
Disable OOV diacritizer	TestSet2	8.8%	2.4%	3.2	1.2
Disable auto corrector	TestSet2	8.3%	2.2%	2.8	1

---

[1] The corpus was revised by a set of linguists who reported these errors.

6   Conclusion

We presented in this paper a hybrid approach for Arabic diacritics restoration. The approach is combining both rule-based and data driven techniques. Our system is trained and tested using the standard ATB corpus for fair comparison with other systems. The system shows improved results over the best reported systems in terms of full form diacritization. As a future work we speculate that further work on POS tagging and disambiguation techniques such as word sense disambiguation could further improve our morphological diacritization.

References

1. Rashwan, M.A.A., et al.: A stochastic arabic diacritizer based on a hybrid of factorized and unfactorized textual features. IEEE Transactions on Audio, Speech, and Language Processing 19, 166–175 (2011)
2. Habash, N., Rambow, O.: Arabic diacritization through full morphological tagging. In: NAACL-Short 2007 Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers, pp. 53–56 (2007)
3. Zitouni, I., Sorensen, J.S., Sarikaya, R.: Maximum entropy based restoration of arabic diacritics. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pp. 577–584 (2006)
4. Habash, N., Rambow, O.: Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL 2005, pp. 573–580 (2005)
5. Buckwalter, T.: Issues in Arabic orthography and morphology analysis. In: Proceedings of the COLING 2004 Workshop on Computational Approaches to Arabic Script-Based Languages, pp. 31–34 (2004)
6. Emam, O., Fisher, V.: A hierarchical approach for the statistical vowelization of arabic text. Tech. rep., IBM (2004)
7. Gimnez, J., Mrquez, L.: Svmtool: A general pos tagging generator based on support vector machines. In: LERC 2004. pp. 573–580 (2004)
8. Maamouri, M., Bies, A., Buckwalter, T., Mekki, W.: The penn arabic treebank: Building a large-scale annotated arabic corpus. In: Arabic Lang. Technol. Resources Int. Conf.; NEMLAR, Cairo, Egypt (2004)
9. Stolcke, A.: Srilman extensible language modeling toolkit. In: Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002), pp. 901–904 (2002)
10. Laerty, J.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: The Eighteenth International Conference on Machine Learning, pp. 282–289 (2001)
11. Jurafsky, D., Martin, J.H.: Speech and Language Processing; an Introduction to Natural Language Processing, Computational Linguistics, and Speech Processing. Prentice-Hall (2000)

EDU-Based Similarity
for Paraphrase Identification

Ngo Xuan Bach, Nguyen Le Minh, and Akira Shimazu

School of Information Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
{bachnx,nguyenml,shimazu}@jaist.ac.jp

Abstract. We propose a new method to compute the similarity between two sentences based on elementary discourse units, EDU-based similarity. Unlike conventional methods, which directly compute similarities based on sentences, our method divides sentences into discourse units and uses them to compute similarities. We also show the relation between paraphrases and discourse units, which plays an important role in paraphrasing. We apply our method to the paraphrase identification task. By using only a single SVM classifier, we achieve 93.1% accuracy on the PAN corpus, a large corpus for detecting paraphrases.

Keywords: Paraphrase Identification, Elementary Discourse Unit, Text Similarity, MT Metrics.

1 Introduction

Paraphrase identification is the task of determining whether two sentences have essentially the same meaning. This task has been shown to play an important role in many natural language applications, including text summarization [4], question answering [15], machine translation [6], and plagiarism detection [34]. For example, detecting paraphrase sentences would help a text summarization system avoid adding redundant information.

Although the paraphrase identification task is defined in the term of semantics, it is usually modeled as a binary classification problem, which can be solved by training a statistical classifier. Many methods have been proposed for identifying paraphrases. These methods usually employ the similarity between two sentences as features, which are computed based on words [10,16,21,25], n-grams [11,21], syntactic parse trees [11,30,33], WordNet [21,25], and MT metrics, the automated metrics for evaluation of translation quality [17,23].

Recently, several studies have shown that discourse structures deliver important information for paraphrase computation. For example, to extract paraphrases, Dolan et al. [14] take the first sentences from comparable documents and consider them as paraphrases. Regneri and Wang [29] introduce a method for collecting paraphrases based on the sequential event order in the discourse.
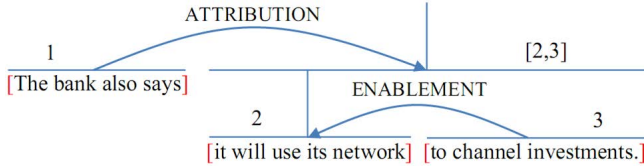
Fig. 1. An example of a discourse tree in RST-DT

However, they only consider some special kinds of data, which the discourse structures can be easily achieved.

Complete discourse structures like in the RST Discourse Treebank (RST-DT) [7] are difficult to achieve though they can be very useful for paraphrase computation [29]. In order to produce such complete discourse structures for a text, we first segment the text into several elementary discourse units (EDUs) (discourse segmentation step). Each EDU may be a simple sentence or a clause in a complex sentence. Consecutive EDUs are then put in relation with each other to create a discourse tree (discourse tree building step) [24]. An example of a discourse tree with three EDUs is shown in Figure 1. Existing full automatic discourse parsing systems are neither robust nor very precise [3,29]. Recently, however, several discourse segmenters with high performance have been introduced [2,19]. The discourse segmenter described in Bach et al. [2] gives 91.0% in the F_1 score on the RST-DT corpus when using Stanford parse trees [20].

In this paper, we present a new method to compute the similarity between two sentences based on elementary discourse units (EDU-based similarity). We first segment two sentences into several EDUs using a discourse segmenter, which is trained on the RST-DT corpus. These EDUs are then employed for computing the similarity between two sentences. The key idea is that for each EDU in one sentence, we try to find the most *similar* EDU in the other sentence and compute the similarity between them. We show how our method can be applied to the paraphrase identification task. Experimental results on the PAN corpus [23] show that our method is effective for the task. To our knowledge, this is the first work that employs discourse units for computing similarity as well as for identifying paraphrases.

The rest of this paper is organized as follows. We first present related work and our contributions in Section 2. Section 3 describes the relation between paraphrases and discourse units. Section 4 presents our method, EDU-based similarity. Experiments on the paraphrase identification task are described in Section 5. Finally, Section 6 concludes the paper.

2   Related Work and Our Contributions

There have been many studies on the paraphrase identification task. Finch et al. [17] use some MT metrics, including BLEU [28], NIST [13], WER [26], and

PER [22] as features for a SVM classifier. Wan et al. [36] combine BLEU features with some others extracted from dependency relations and tree edit-distance. They also take SVMs as the learning method to train a binary classifier. Mihalcea et al. [25] use pointwise mutual information, latent semantic analysis, and WordNet to compute an arbitrary text-to-text similarity metric. Kozareva and Montoyo [21] employ features based on longest common subsequence (LSC), skip n-grams, and WordNet. They use a meta-classifier composed of SVMs, k-nearest neighbor, and maximum entropy models. Rus et al. [30] adapt a graph-based approach (originally developed for recognizing textual entailment) for paraphrase identification. Fernando and Stevenson [16] build a matrix of word similarities between all pairs of words in both sentences. Das and Smith [11] introduce a probabilistic model which incorporates both syntax and lexical semantics using quasi-synchronous dependency grammars for identifying paraphrases. Socher et al. [33] describe a joint model that uses the features extracted from both single words and phrases in the parse trees of the two sentences.

Most recently, Madnani et al. [23] present an investigation of the impact of MT metrics on the paraphrase identification task. They examine 8 different MT metrics, including BLEU [28], NIST [13], TER [31], TERP [32], METEOR [12], SEPIA [18], BADGER [27], and MAXSIM [8], and show that a system using nothing but some MT metrics can achieve state-of-the-art results on this task. In our work, we also employ MT metrics as features of a paraphrase identification system. The method of using them, however, is very different from the method in previous work.

Discourse structures have only marginally been considered for paraphrase computation. Regneri and Wang [29] introduce a method for collecting paraphrases using discourse information on a special type of data, TV show episodes. With such kind of data, they assume that discourse structures can be achieved by taking sentence sequences of recaps. Our work employ the recent advances in discourse segmentation. Hernault et al. [19] present a sequence model for segmenting texts into discourse units using Conditional Random Fields. Bach et al. [2] introduce a reranking model for discourse segmentation using subtree features. Two segmenters achieve 89.0% and 91.0%, respectively, in the F_1 score on RST-DT when using Stanford parse trees.

The aim of our work is to exploit discourse information for computing paraphrases in general texts. Our main contributions can be summarized in the following points:

1. We show the relation between discourse units and paraphrasing, in which discourse units play an important role in paraphrasing.
2. We present EDU-based similarity, a new method for computing the similarity between two sentences based on elementary discourse units.
3. We apply the method to the task of paraphrase identification.
4. We conduct experiments on the PAN corpus [23] to show that EDU-based similarity is effective for the task of identifying paraphrases.

[Or his needful holiday has come,]$_{1A}$ [and he is staying at a friend's house,]$_{1B}$ [or is thrown into new intercourse at some health-resort.]$_{1C}$

[Or need a holiday has come,]$_{2A}$ [and he stayed in the house of a friend]$_{2B}$ [or disposed of in a new relationship to a health resort.]$_{2C}$
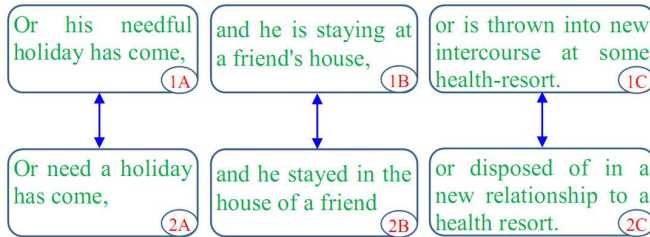
Fig. 2. A paraphrase sentence pair in the PAN corpus [23]

3   Paraphrases and Discourse Units

In this section, we describe the relation between paraphrases and discourse units. We will show that discourse units are blocks which play an important role in paraphrasing.

Figure 2 shows an example of a paraphrase sentence pair. In this example, the first sentence can be divided into three elementary discourse units (EDUs), 1A, 1B, and 1C, and the second sentence can also be segmented into three EDUs, 2A, 2B, and 2C. Comparing these six EDUs, we can see that they make three aligned pairs of paraphrases: 1A with 2A, 1B with 2B, and 1C with 2C. Therefore, if we consider the first sentence is the original sentence, the second sentence can be created by paraphrasing each discourse unit in the original sentence.

Figure 3 shows a more complex case. The first sentence consists of four EDUs, 3A, 3B, 3C, and 3D; and the second sentence includes four EDUs, 4A, 4B, 4C, and 4D. In this case, if we consider the first sentence is the original one, we have some remarks:

- The discourse unit 4A is a paraphrase of the discourse unit 3B,
- The unit 4B is a paraphrase of the combination of two units, 3A and 3C, and
- The combination of two units 4C and 4D is a paraphrase of the unit 3D.

By analyzing paraphrase sentences, we found that discourse units are very important to paraphrasing. In many cases, a paraphrase sentence can be created by applying the following operations to the original sentence:

1. Reordering two discourse units,
2. Combining two discourse units into one unit,
3. Dividing one discourse unit into two units, and
4. Paraphrasing a discourse unit.

[Age of consent legislation,]$_{3A}$ [as applied to the question of social vice,]$_{3B}$ [is one thing,]$_{3C}$ [and consent as applied to the question of slavery , quite another thing.]$_{3D}$

[When applied to social vices,]$_{4A}$ [age of consent legislation is one thing,]$_{4B}$ [when the legislation is applied to slavery,]$_{4C}$ [a totally different and epidemic problem exists.]$_{4D}$
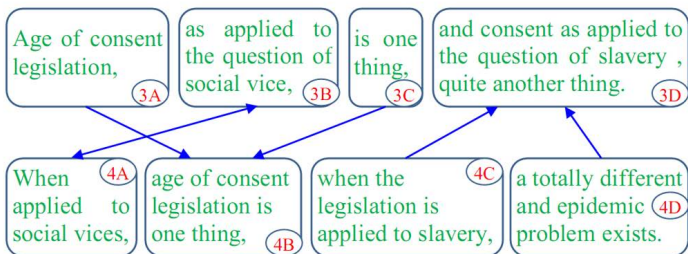
Fig. 3. Another paraphrase sentence pair in the PAN corpus

An example of Operation 1 and Operation 2 is the case of units 3A, 3B, and 3C in Figure 3 (reordering 3A and 3B, and then combining 3A and 3C). Unit 3D illustrates an example for Operation 3. The last operation is the most important operation, which is applied to almost all of discourse units.

4   EDU-Based Similarity

Motivated from the analysis of the relation between paraphrases and discourse units, we propose a method to compute the similarity between two sentences. Our method considers each sentence as a sequence of EDUs.

First, we present the notion of *ordered similarity functions*. Given two arbitrary texts t_1 and t_2, an ordered similarity function $Sim_{ordered}(t_1, t_2)$ will return a real score, which measures how t_1 is similar to t_2. Note that in this function, the roles of t_1 and t_2 are different, in which t_2 can be seen as a *gold* standard and we want to evaluate t_1 based on t_2. Examples of ordered similarity functions are MT metrics, which evaluate how a hypothesis text (t_1) is similar to a reference text (t_2).

Given an ordered similarity function $Sim_{ordered}$, we can define the similarity between two arbitrary texts t_1 and t_2 as follows:

$$Sim(t_1, t_2) = \frac{Sim_{ordered}(t_1, t_2) + Sim_{ordered}(t_2, t_1)}{2}. \tag{1}$$

Let (s_1, s_2) be a sentence pair, then s_1 and s_2 can be represented as sequences of elementary discourse units: $s_1 = (e_1, e_2, \ldots, e_m)$ and $s_2 = (f_1, f_2, \ldots, f_n)$, where m and n are the numbers of discourse units in s_1 and s_2, respectively. We define an ordered similarity function between s_1 and s_2 as follows:

$$Sim_{ordered}(s_1, s_2) = \sum_{i=1}^{m} Imp(e_i, s_1) * Sim_{ordered}(e_i, s_2) \tag{2}$$

where $Imp(e_i, s_1)$ is the importance of the discourse unit e_i in the sentence s_1, and $Sim_{ordered}(e_i, s_2)$ is the ordered similarity between the discourse unit e_i and the sentence s_2.

In this work, we simply consider that all words contribute equally to the meaning of the sentence. Therefore, the importance function can be computed as follows:

$$Imp(e_i, s_1) = \frac{|e_i|}{|s_1|} \tag{3}$$

where $|e_i|$ and $|s_1|$ are the lengths (in words) of the discourse unit e_i and the sentence s_1, respectively.

The ordered similarity $Sim_{ordered}(e_i, s_2)$ is computed based on the discourse unit f_j in the sentence s_2, which is the most similar to e_i:

$$Sim_{ordered}(e_i, s_2) = Max_{j=1}^n Sim_{ordered}(e_i, f_j). \tag{4}$$

Substituting (3) and (4) into (2) we have:

$$Sim_{ordered}(s_1, s_2) = \sum_{i=1}^m \frac{|e_i|}{|s_1|} Max_{j=1}^n Sim_{ordered}(e_i, f_j). \tag{5}$$

Finally, from (5) and (1) we have the formula for computing the similarity between two sentences based on their discourse units (EDU-based similarity), where the ordered similarity between two units is computed directly using the definition of the ordered similarity function, as follows:

$$\begin{aligned} Sim(s_1, s_2) &= \frac{Sim_{ordered}(s_1, s_2) + Sim_{ordered}(s_2, s_1)}{2} \\ &= \frac{1}{2} * \sum_{i=1}^m \frac{|e_i|}{|s_1|} * Max_{j=1}^n Sim_{ordered}(e_i, f_j) \\ &+ \frac{1}{2} * \sum_{j=1}^n \frac{|f_j|}{|s_2|} * Max_{i=1}^m Sim_{ordered}(f_j, e_i). \end{aligned} \tag{6}$$

We now present an example of computing the EDU-based similarity between two sentences in Figure 2 using the BLEU score. Table 1 shows the basic information of the calculation step by step. Line 1 and line 2 present two tokenized sentences and their lengths in words. Lines 3 through 5 compute the similarity between two sentences directly based on sentences. By using this method, the similarity is 0.5332. Elementary discourse units of two sentences are shown in lines 6 through 11. The computation of EDU-based similarity is described in lines 12 through 20. By using this method, the similarity is 0.5369, which is slightly higher than the similarity computed directly using sentences.

5   Experiments

This section describes our experiments on the paraphrase identification task using EDU-based similarities as features for an SVM classifier [35]. Like the

Table 1. An example of computing sentence-based and EDU-based similarities

| Line	Computation		
1	s_1: Or his needful holiday has come , and he is staying at a friend 's house , or is thrown into new intercourse at some health-resort .	Length=27	
2	s_2: Or need a holiday has come , and he stayed in the house of a friend , or disposed of in a new relationship to a health resort .	Length=29	
	Sentence-based Similarity		
3	BLEU$(s_1, s_2) = $ **0.5333**		
4	BLEU$(s_2, s_1) = $ **0.5330**		
5	$Sim(s_1, s_2) = \frac{BLEU(s_1,s_2)+BLEU(s_2,s_1)}{2} = $ **0.5332**		
	Discourse Units		
6	e_1: Or his needful holiday has come ,	Length=7	
7	e_2: and he is staying at a friend 's house ,	Length=10	
8	e_3: or is thrown into new intercourse at some health-resort .	Length=10	
9	f_1: Or need a holiday has come ,	Length=7	
10	f_2: and he stayed in the house of a friend ,	Length= 10	
11	f_3: or disposed of in a new relationship to a health resort .	Length=12	
	EDU-based Similarity		
12	BLEU$(e_1, f_1) = $ **0.7143**	BLEU$(e_1, f_2) = 0.0931$	BLEU$(e_1, f_3) = 0.0699$
13	BLEU$(e_2, f_1) = 0.1818$	BLEU$(e_2, f_2) = $ **0.5455**	BLEU$(e_2, f_3) = 0.0830$
14	BLEU$(e_3, f_1) = 0.0833$	BLEU$(e_3, f_2) = 0$	BLEU$(e_3, f_3) = $ **0.4167**
15	EDU_BLEU$(s_1, s_2) = \frac{7}{27} * 0.7143 + \frac{10}{27} * 0.5455 + \frac{10}{27} * 0.4167 = $ **0.5416**		
16	BLEU$(f_1, e_1) = $ **0.7143**	BLEU$(f_1, e_2) = 0.1613$	BLEU$(f_1, e_3) = 0.0699$
17	BLEU$(f_2, e_1) = 0.1000$	BLEU$(f_2, e_2) = $ **0.5429**	BLEU$(f_2, e_3) = 0$
18	BLEU$(f_3, e_1) = 0.0833$	BLEU$(f_3, e_2) = 0.0833$	BLEU$(f_3, e_3) = $ **0.4167**
19	EDU_BLEU$(s_2, s_1) = \frac{7}{29} * 0.7143 + \frac{10}{29} * 0.5429 + \frac{12}{29} * 0.4167 = $ **0.5321**		
20	EDU_Sim$(s_1, s_2) = \frac{EDU_BLEU(s_1,s_2)+EDU_BLEU(s_2,s_1)}{2} = $ **0.5369**		

work of Madnani et al. [23], we employed MT metrics as the ordered similarity functions. However, we computed MT metrics based on EDUs in addition to MT metrics based on sentences. To segment sentences, we implemented the discourse segmenter described in Bach et al. [2]. In all experiments, parse trees were obtained by using the Stanford parser [20].

5.1   Data and Evaluation Method

We conducted experiments on the PAN corpus, a corpus for paraphrase identification task created from a plagiarism detection corpus [23]. Table 2 shows statistics on the corpus. The corpus includes a training set of $10,000$ sentence pairs and a test set of $3,000$ sentence pairs. On average, each sentence contains

Table 2. PAN corpus for paraphrase identification

	Training Set	Test Set
Number of sentence pairs	10,000	3,000
Number of EDUs per sentence	4.31	4.33
Number of words per sentence	40.07	41.12

about 4.3 discourse units, and about 40.1 words in the training set, 41.1 words in the test set. We chose this corpus for these reasons. First, it is a large corpus for detecting paraphrases. Second, it contains many long sentences. Our method computes similarities based on discourse units. It is suitable for long sentences with several EDUs. Last, according to Madnani et al. [23], the PAN corpus contains many realistic examples of paraphrases.

We evaluated the performance of our paraphrase identification system by accuracy and the F_1 score. The accuracy was the percentage of correct predictions over all the test set, while the F_1 score was computed only based on the paraphrase sentence pairs[1].

5.2   MT Metrics

We investigated our method with six different MT metrics (six types of ordered similarity functions). These metrics have been shown to be effective for the task of paraphrase identification [23].

1. BLEU [28] is the most commonly used MT metric. It computes the amount of n-gram overlap between a hypothesis text (the output of a translation system) and a reference text.
2. NIST [13] is a variant of BLEU using the arithmetic mean of n-gram overlaps. Both BLEU and NIST use exact matching. They have no concept of synonymy or paraphrasing.
3. TER [31] computes the number of edits needed to "fix" the hypothesis text so that it matches the reference text.
4. TERP [32] or TER-Plus is an extension of TER, that utilizes phrasal substitutions, stemming, synonyms, and other improvements.
5. METEOR [12] is based on the harmonic mean of unigram precision and recall. It also incorporates stemming, synonymy, and paraphrase.
6. BADGER [27], a language independent metric, computes a compression distance between two sentences using the Burrows Wheeler Transformation (BWT).

Among six MT metrics, TER and TERP compute a translation error rate between a hypothesis text and a reference text. Therefore, the smaller they are, the more similar the two texts are. When using these metrics in computing EDU-based similarities, we replaced the *max* function in Equation (6) by a *min* function.

---

[1] If we consider each sentence pair as an instance with label +1 for *paraphrase* and label -1 for *non-paraphrase*, the reported F_1 score was the F_1 score on label +1.

Table 3. Experimental results on each individual MT metric

MT Metric	Sentence-based similarities		+ EDU-based similarities	
	Accuracy(%)	F$_1$(%)	Accuracy(%)	F$_1$(%)
BLEU(1-4)	89.0	88.4	**89.6(+0.6)**	89.1(+0.7)
NIST(1-5)	84.6	82.7	**87.6(+3.0)**	86.8(+4.1)
TER	88.2	87.3	**88.5(+0.3)**	87.7(+0.4)
TERP	91.0	90.6	91.1(+0.1)	90.8(+0.2)
METEOR	90.0	89.6	89.8(-0.2)	89.4(-0.2)
BADGER	88.1	87.8	88.2(+0.1)	87.8(-)

5.3 Experimental Results

In all experiments, we chose SVMs [35] as the learning method to train a binary classifier[2].

First, we investigated each individual MT metric. To see the contributions of EDU-based similarities, we conducted experiments in two settings. In the first setting, we directly applied the MT metric to pairs of sentences to get the similarities (sentence-based similarities). In the second one, we computed EDU-based similarities in addition to the sentence-based similarities. Like Madnani et al. [23], in our experiments, we used BLEU1 through BLEU4 as 4 different features and NIST1 through NIST5 as 5 different features[3]. Table 3 shows experimental results in two settings on the PAN corpus. We can see that, adding EDU-based similarities improved the performance of the paraphrase identification system with most of the MT metrics, especially with NIST(3.0%), BLEU (0.6%), and TER (0.3%).

Table 4 shows experimental results with multiple MT metrics on the PAN corpus. With each MT metric, we computed the similarities in both methods, based directly on sentences and based on discourse units. We gradually added MT metrics one by one to the system. After adding the TERP metric, we achieved 93.1% accuracy and 93.0% in the F$_1$ score. Adding more two metrics METEOR and BADGER, the performance was not improved.

Two last rows of Table 4 shows the results of Madnani et al. [23] when using 4 MT metrics, including BLEU, NIST, TER, and TERP (Madnani-4) and when using all 6 MT metrics (Madnani-6)[4]. Compared with the best previous results, our method improves 0.8% accuracy and 0.9% in the F$_1$ score. It yields a 10.4% error rate reduction. Also note that, the previous work employs a meta-classifier with three constituent classifiers, Logistic regression, SVMs, and instance-based learning, while we use only a single classifier with SVMs.

We also investigated our method on long and short sentences. We divided sentence pairs in the test set into two subsets: Subset1 (long sentences) contains

---

[2] We conducted experiments on LIBSVM tool [9] with the RBF kernel.

[3] BLEUn and NISTn use n-grams.

[4] Madnani et al. [23] show that adding more MT metrics does not improve the performance of the paraphrase identification system.

Table 4. Experimental results on combined MT metrics

MT Metrics	Accuracy(%)	F_1(%)
BLEU	89.6	89.1
BLEU+NIST	91.2	90.9
BLEU+NIST+TER	91.8	91.6
BLEU+NIST+TER+TERP	**93.1**	**93.0**
Madnani-4	91.5	91.2
Madnani-6	92.3	92.1

Table 5. Experimental results on long and short sentences

Subset	#sent pairs	#EDUs/sent	#words/sent	Acc.(%)	F_1(%)
Subset1	1317	6.5	56.6	**96.6**	**94.8**
Subset2	1683	2.6	27.2	90.4	92.3

sentence pairs that both sentences have at least 4 discourse units[5], and Subset2 (short sentences) contains the other sentence pairs. Table 5 shows the information and experimental results on two subsets. Subset1 consists of 1317 sentence pairs (on average, 6.5 EDUs and 56.6 words per sentence), while Subset2 consists of 1683 sentence pairs (on average, 2.6 EDUs and 27.2 words per sentence). We can see that, our method was effective for the long sentences, which we achieved 96.6% accuracy and 94.8% in the F_1 score compared with 90.4% accuracy and 92.3% in the F_1 score of the short sentences.

6   Conclusion

In this paper, we proposed a new method to compute the similarity between two sentences based on elementary discourse units, EDU-based similarity. This method was motivated from the analysis of the relation between paraphrases and discourse units. By analyzing examples of paraphrases, we found that discourse units play an important role in paraphrasing. We applied EDU-based similarity to the task of paraphrase identification. Experimental results on the PAN corpus showed the effectiveness of the proposed method. To the best of our knowledge, this is the first work to employ discourse units for computing similarity as well as for identifying paraphrases. Although our method is proposed for computing the similarity between two sentences, it can be also used to compute the similarity between two arbitrary texts.

In the future, we would like to apply our method to other datasets for the paraphrase identification task as well as to other related tasks such as recognizing textual entailment [5] and semantic textual similarity [1]. Another direction is to improve the method of computing similarity, especially how to evaluate the

---

[5] Number 4 was chosen because on average each sentence contains about 4 EDUs (see Table 2).

importance of a discourse unit in a sentence. In this work, we simply consider that discourse units are independent and all words contribute equally to the meaning of the sentence. Therefore, the importance of discourse units is only calculated based on their lengths (in words). Exploiting the relations between discourse units for computing similarity may be an interesting direction.

References

1. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A.: SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In: Proceedings of SemEval, pp. 385–393 (2012)
2. Bach, N.X., Minh, N.L., Shimazu, A.: A Reranking Model for Discourse Segmentation using Subtree Features. In: Proceedings of SIGDIAL, pp. 160–168 (2012)
3. Bach, N.X., Le Minh, N., Shimazu, A.: UDRST: A Novel System for Unlabeled Discourse Parsing in the RST Framework. In: Isahara, H., Kanzaki, K. (eds.) JapTAL 2012. LNCS (LNAI), vol. 7614, pp. 250–261. Springer, Heidelberg (2012)
4. Barzilay, R., McKeown, K.R., Elhadad, M.: Information Fusion in the Context of Multi-Document Summarization. In: Proceedings of ACL, pp. 550–557 (1999)
5. Bentivogli, L., Dagan, I., Dang, H.T., Giampiccolo, D., Magnini, B.: The fifth Pascal Recognizing Textual Entailment Challenge. In: Proceedings of TAC (2009)
6. Callison-Burch, C., Koehn, P., Osborne, M.: Improved Statistical Machine Translation Using Paraphrases. In: Proceedings of NAACL, pp. 17–24 (2006)
7. Carlson, L., Marcu, D., Okurowski, M.E.: RST Discourse Treebank. Linguistic Data Consortium (LDC) (2002)
8. Chan, Y.S., Ng, H.T.: MAXSIM: A Maximum Similarity Metric for Machine Translation Evaluation. In: Proceedings of ACL-HLT, pp. 55–62 (2008)
9. Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology 2(3), 27:1-27:27 (2011)
10. Corley, C., Mihalcea, R.: Measuring the Semantic Similarity of Texts. In: Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, pp. 13–18 (2005)
11. Das, D., Smith, N.A.: Paraphrase Identification as Probabilistic Quasi-Synchronous Recognition. In: Proceedings of ACL-IJCNLP, pp. 468–476 (2009)
12. Denkowski, M., Lavie, M.: Extending the METEOR Machine Translation Metric to the Phrase Level. In: Proceedings of NAACL, pp. 250–253 (2010)
13. Doddington, G.: Automatic Evaluation of Machine Translation Quality using N-gram Co-occurrence Statistics. In: Proceedings of the 2nd International Conference on Human Language Technology Research, pp. 138–145 (2002)
14. Dolan, B., Quirk, C., Brockett, C.: Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In: Proceedings of COLING, pp. 350–356 (2004)
15. Duboue, P.A., Chu-Carroll, J.: Answering the Question You Wish They had Asked: The Impact of Paraphrasing for Question Answering. In: Proceedings of NAACL, pp. 33–36 (2006)
16. Fernando, S., Stevenson, M.: A Semantic Similarity Approach to Paraphrase Detection. In: Proceedings of CLUK (2008)

17. Finch, A., Hwang, Y.S., Sumita, E.: Using Machine Translation Evaluation Techniques to Determine Sentence-level Semantic Equivalence. In: Proceedings of the 3rd International Workshop on Paraphrasing, pp. 17–24 (2005)
18. Habash, N., Kholy, A.E.: SEPIA: Surface Span Extension to Syntactic Dependency Precision-based MT Evaluation. In: Proceedings of the Workshop on Metrics for Machine Translation at AMTA (2008)
19. Hernault, H., Bollegala, D., Ishizuka, M.: A Sequential Model for Discourse Segmentation. In: Gelbukh, A. (ed.) CICLing 2010. LNCS, vol. 6008, pp. 315–326. Springer, Heidelberg (2010)
20. Klein, D., Manning, C.: Accurate Unlexicalized Parsing. In: Proceedings of ACL, pp. 423–430 (2003)
21. Kozareva, Z., Montoyo, A.: Paraphrase Identification on the Basis of Supervised Machine Learning Techniques. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) FinTAL 2006. LNCS (LNAI), vol. 4139, pp. 524–533. Springer, Heidelberg (2006)
22. Leusch, G., Ueffing, N., Ney, H.: A Novel String-to-String Distance Measure with Applications to Machine Translation Evaluation. In: Proceedings of MT Summit IX (2003)
23. Madnani, N., Tetreault, J., Chodorow, M.: Re-examining Machine Translation Metrics for Paraphrase Identification. In: Proceedings of NAACL-HLT, pp. 182–190 (2012)
24. Mann, W.C., Thompson, S.A.: Rhetorical Structure Theory. Toward a Functional Theory of Text Organization. Text 8, 243–281 (1988)
25. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In: Proceedings of AAAI, pp. 775–780 (2006)
26. Niessen, S., Och, F.J., Leusch, G., Ney., H.: An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. In: Proceedings of LREC (2000)
27. Parker, S.: BADGER: A New Machine Translation Metric. In: Proceedings of the Workshop on Metrics for Machine Translation at AMTA (2008)
28. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A Method for Automatic Evaluation of Machine Translation. In: Proceedings of ACL, pp. 311–318 (2002)
29. Regneri, M., Wang, R.: Using Discourse Information for Paraphrase Extraction. In: Proceedings of EMNLP-CONLL, pp. 916–927 (2012)
30. Rus, V., McCarthy, P.M., Lintean, M.C., McNamara, D.S., Graesser, A.C.: Paraphrase Identification with Lexico-Syntactic Graph Subsumption. In: Proceedings of FLAIRS Conference, pp. 201–206 (2008)
31. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A Study of Translation Edit Rate with Targeted Human Annotation. In: Proceedings of the Conference of the Association for Machine Translation in the Americas, AMTA (2006)
32. Snover, M., Madnani, N., Dorr, B., Schwartz, R.: TER-Plus: Paraphrase, Semantic, and Alignment Enhancements to Translation Edit Rate. Machine Translation 23(23), 117–127 (2009)
33. Socher, R., Huang, E.H., Pennington, J., Ng, A.Y., Manning, C.D.: Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In: Advances in Neural Information Processing Systems 24 (NIPS), pp. 801–809 (2011)
34. Uzuner, O., Katz, B., Nahnsen, T.: Using Syntactic Information to Identify Plagiarism. In: Proceedings of the 2nd Workshop on Building Educational Applications using Natural Language Processing, pp. 37–44 (2005)
35. Vapnik, V.N.: Statistical Learning Theory. Wiley Interscience (1998)
36. Wan, S., Dras, R., Dale, M., Paris, C.: Using Dependency-Based Features to Take the "Para-farce" out of Paraphrase. In: Proceedings of the 2006 Australasian Language Technology Workshop, pp. 131–138 (2006)

Exploiting Query Logs and Field-Based Models to Address Term Mismatch in an HIV/AIDS FAQ Retrieval System

Edwin Thuma, Simon Rogers, and Iadh Ounis

School of Computing Science,
University of Glasgow,Glasgow, G12 8QQ, UK
`thumae@dcs.gla.ac.uk,{simon.rogers,iadh.ounis}@glasgow.ac.uk`

Abstract. One of the main challenges in the retrieval of Frequently Asked Questions (FAQ) is that the terms used by information seekers to express their information need are often different from those used in the relevant FAQ documents. This lexical disagreement (aka term mismatch) can result in a less effective ranking of the relevant FAQ documents by retrieval systems that rely on keyword matching in their weighting models. In this paper, we tackle such a lexical gap in an SMS-Based HIV/AIDS FAQ retrieval system by enriching the traditional FAQ document representation using terms from a query log, which are added as a separate field in a field-based model. We evaluate our approach using a collection of FAQ documents produced by a national health service and a corresponding query log collected over a period of 3 months. Our results suggest that by enriching the FAQ documents with additional terms from the SMS queries for which the true relevant FAQ documents are known and combining term frequencies from the different fields, the lexical mismatch problem in our system is markedly alleviated, leading to an overall improvement in the retrieval performance in terms of Mean Reciprocal Rank (MRR) and recall.

Keywords: Frequently Asked Question, Term Mismatch, Query Logs, Field-Based Model.

1 Introduction

We have developed an Automated SMS-Based HIV/AIDS FAQ retrieval system that can be queried by users to provide answers on HIV/AIDS related questions. The system uses, as its information source, the full HIV/AIDS FAQ question-answer booklet provided by the Ministry of Health (MOH) in Botswana for its IPOLETSE[1] call centre. This FAQ question-answer booklet is made up of 205 question-answer pairs organised into eleven chapters of varying sizes. For example, there is a chapter on "Nutrition, Vitamins and HIV/AIDS" and a chapter on "Men and HIV/AIDS". Below is an example of a question-answer pair entry that can be found in Chapter Eight, "Introduction to ARV Therapy":

---

[1] `http://www.hiv.gov.bw/content/ipoletse`

Question : What is the importance of taking ARV therapy if there is no cure for AIDS?

Answer : Although ARV therapy is not a cure for AIDS, it enables you to live a longer and more productive life if you take it the right way. ARV therapy is just like treatment for chronic illnesses such as diabetes or high blood pressure.

For the remainder of this paper, we will refer to a question-answer pair as the FAQ document and the set of all 205 FAQ documents as the FAQ document collection. The users' SMS messages will be referred to as queries.

One key problem in this domain is that there will often be term mismatch between the queries from the users and the relevant FAQ documents [18,19]. For example, the user's query: *"Is HIV/AIDS gender based to some extent?"* and the FAQ document: *"Does HIV/AIDS affect women differently from men? No, the virus affects both men and women in exactly the same way i.e. by making the immune system weak, so that it cannot fight off other illnesses"* are semantically similar but lexically different. This term mismatch between the user's query and the relevant FAQ document may result in a less effective ranking by a retrieval system that relies on keywords matching in its weighting model [3].

To solve this term mismatch problem between the users' queries and the relevant FAQ documents in the FAQ document collection, query log clustering is often used [6]. Earlier work by Kim et al. [6,7] suggests that a good clustering of query logs can markedly reduce the term mismatch problem that arises in an FAQ retrieval system, thus improving the overall retrieval performance. Another approach that is often used in the Information Retrieval (IR) community to alleviate the term mismatch problem is query expansion. Various authors have reported mixed results [3,20]. For example, Voorhees [20] did not show any significant improvement if queries are expanded with terms from WordNet. On the other-hand, Fang [3] has shown significant performance improvement when hand-crafted lexical resources are used for query expansion.

In this paper, we aim to tackle this term mismatch problem in an SMS-based HIV/AIDS FAQ retrieval system by enriching the traditional FAQ document representation (Question and Answer) using terms from a query log, which are added as a separate field in a field-based model [10,16]. Our main contribution is to demonstrate that enriching the FAQ documents (Question and Answer Fields only) with additional terms from potential users of the FAQ system can alleviate the term mismatch problem that arises in our FAQ retrieval system. This will be measured by an increase in recall. Recall is the fraction of relevant documents to the query that are retrieved. We thoroughly evaluate our approach using the aforementioned HIV/AIDS question-answer booklet provided by the Ministry of Health in Botswana as our information source and a corresponding query log collected in Botswana over a period of 3 months.

The rest of this paper is organised as follows: In Section 2 we survey related work, followed by a description of our enrichment strategies in Section 3. In Section 4 we describe how the SMS queries were collected and analysed. Then we describe our experimental setting in Section 5, followed by the experimental results in Section 6 and the conclusions in Section 7.

2   Related Work

Earlier FAQ retrieval systems [4,18,21] relied on knowledge bases to alleviate term mismatch between the query and the relevant FAQ documents. For example, in the system proposed by Sneiders [18], each FAQ is analysed and annotated with three keywords types: required keywords, optional keywords and irrelevant keywords. For each user query, the system retrieves and ranks the relevant FAQs according to the three keyword types. The system rejects the match between the user's query and an FAQ document in the collection if there is at least one required keyword missing in the user's query. It is worth noting that these early representative systems rely on knowledge bases that require a lot of time to construct whenever new FAQs are added to the collection or the application domain changes.

Jeon et al. [5] and Xue et al. [22] proposed a translation based retrieval model that uses the similarity between answers of lexically different but semantically similar questions in community based question-answer archives to learn translation probabilities. They used the learned translation probabilities to search semantically similar questions and their results suggest that their approach outperforms other baseline retrieval models: the vector space model with cosine similarity, the Okapi BM25 model and the query-likelihood language model. The approach proposed by Jeon et al. and Xue et al. shows promising results for a large collection of question-answer archives. However, their approach may not work in our HIV/AIDS FAQ retrieval system because it uses a small fixed dataset of question-answer pairs (205). Learning good translation probabilities might be difficult for such a small dataset.

Kim et al. [6] on the other-hand proposed a more adaptable approach that uses query logs as knowledge sources to solve the term mismatch problem in an FAQ retrieval system. Their system called FRACT is made up of two subsystems, a query log clustering system and a cluster based retrieval system. The query log clustering system considers each FAQ as an independent category and it periodically collects and refines the users' query logs that are then classified into each FAQ category by using a vector similarity in the latent semantic space. FRACT uses the clustered query logs to associate every users' question to the relevant cluster of FAQs and ranks and return a list of FAQs based on the similarity with the cluster.

More recently Moreo et al. [11], introduced a new method called Minimal Differentiator Expression (MDE). In their approach, they solve the term mismatch problem by using linguistic classifiers that they trained using expressions that totally differentiate each FAQ. They enhance the performance of their system during the life of its operation by continuously training the classifier with new evidence from the users' queries. Their approach although different from our proposed approach also relies on query logs to resolve the term mismatch problem. In their evaluation, they reported that their approach outperformed the cluster based retrieval proposed in [6]. Other approaches that closely resemble our work

are the document expansion approach proposed in [2,17] and the query expansion approach in [1]. The document expansion approach proposed by Billerbeck and Zobel [2] yielded unpromising results and this might be partly due to the fact that the expansion terms were selected automatically without using the actual query relevance judgements. Hence this might have resulted in the wrong terms being used to expand irrelevant documents. In this work, we will rely on the query relevance judgements to avoid linking query terms to irrelevant FAQ documents.

3   FAQ Documents Enrichment Strategies

In Web IR, there is the notion of document fields and this provides a way to incorporate the structure of a document in the retrieval process [16]. For example, the contents of different HTML tags (e.g anchor text,title, body) are often used to represent different document fields [13,16]. Earlier work by [10] has shown that combining evidence from different fields in Web retrieval improves retrieval performance. In this paper, we represent the FAQ document made up of question-answer pairs into a $QUESTION$ and an $ANSWER$ field. We then introduce a third field, $FAQLog$, that we use to add additional terms from queries for which the true relevant FAQ documents are known. We aim to solve the term mismatch problem in our FAQ retrieval system by combining evidence from these three fields.

We will evaluate the proposed approach using two different enrichment strategies. First, we enrich the FAQ documents using all the terms from a query log. In this approach, all the queries from the training set for which the true relevant FAQ documents are known will be added into the new introduced $FAQLog$ field as shown in Table 1. In other words, if an FAQ document is known to be relevant to a query, then this query is added to its $FAQLog$ field. For the remainder of this paper we will refer to this approach as the Term Frequency approach. In the second approach, we will enrich the FAQ documents using term occurrences from a query log. Here, all the unique terms from the training set for which the true relevant FAQ documents are known will be added to the $FAQLog$ field as shown in Table 2. In other words, only new query terms that do not appear in the $FAQLog$ field will be added to that field. For the remainder of this paper we will refer to this approach as the Term Occurrence approach. We will apply field-based weighting models on the enriched FAQ documents using PL2F [10] and BM25F [16].

The main difference between the two enrichment approaches is that the frequencies with which users use some rare terms in specific FAQ documents can be captured if the term frequency enrichment approach is used. For example, under the term frequency approach (Table 1), the term frequencies of the terms *gender* and *infected* in the FAQLog field are: $gender = 2$ and $infected = 2$. Under the term occurrence approach (Table 2) the term frequencies of these terms are 1 because the query terms under this approach can only be added to this field once even if they appear in many queries. Since, both BM25F and PL2F rely on term frequencies to calculate the final retrieval score of a relevant document given a

Table 1. Enrichment Using Query Term Frequencies

FIELDS	CONTENTS of FIELDS
QUESTION	Does HIV / AIDS affect women differently from men?
ANSWER	No, the virus affects both men and women in exactly the same way i.e. by making the immune system weak, so that it cannot fight off other illnesses.
FAQLog	Is HIV/AIDS gender based to some extent? Between men and women, who are most infected by HIV/AIDS? who are mainly infected male or female? which gender is mostly affected by the disease?

Table 2. Enrichment Using Query Term Occurrence

FIELDS	CONTENTS of FIELDS
QUESTION	Does HIV / AIDS affect women differently from men?
ANSWER	No, the virus affects both men and women in exactly the same way i.e. by making the immune system weak, so that it cannot fight off other illnesses.
FAQLog	is, hiv, aids, gender, based, to, some, extent, between, men, and, women, who, are, most, infected, by, mainly, male, or, female, which, mostly, affected, the, disease

query, our two enrichment strategies will always give different retrieval scores. We will investigate the usefulness of each enrichment approach in Section 5.

4   Collecting and Analysing SMS Queries

85 participants were recruited in Botswana and asked to provide SMS queries on the general topic of HIV/AIDS. Having provided SMS queries, they then used a web-based interface to find the relevant FAQ documents from the FAQ document collection using the SMS queries. This provided us with SMS queries linked to the appropriate FAQ documents in the collection. In total, 957 SMS queries were collected of which 750 could be matched to an FAQ document in the collection. The remaining 207 did not match anything in the collection and investigating how to detect such orphan queries in a real system is a subject for future work. The 750 SMS queries that could be matched spanned 131 of the 205 FAQ documents, leaving 74 FAQ documents with no SMS queries.

We analysed these SMS queries, counting the number of queries that matched each FAQ document. Our analysis shows that the distribution of queries per FAQ document was not spread evenly. There were some FAQ documents that matched more than 20 users' queries. This was more evident on a topic related to the prevention and transmission of HIV and AIDS. Similar findings were also reported by Sneiders [19] who concluded that people who share the same interest tend to ask the same question over and over again. In this paper, we exploit this repetitive nature of the query log by proposing to enrich the FAQ documents with SMS queries for which the true relevant FAQ document is known thus reducing the term mismatch problem in our FAQ retrieval system.

5   Experimental Description

We begin Section 5.1 by describing our experimental settings followed by a description of our experimental investigations and our baseline systems in Section 5.2. We then describe how we created the new enriched FAQ document representation with the query logs followed by a description of how the field weights for the field-based weighting models were optimised in Section 5.3.

5.1 Experimental Setting

For all our experimental evaluation, we used the Terrier-3.5[2] [12], an open source Information Retrieval (IR) platform. All the FAQ documents used in this study were first pre-processed before indexing and this involved tokenising the text and stemming each token using the full Porter [14] stemming algorithm. To filter out terms that appear in a lot of FAQ documents, we did not use a stopword list during the indexing and the retrieval process. Instead, we ignored the terms that had low Inverse Document Frequency (IDF) when scoring the documents. Indeed, all the terms with term frequency higher than the number of the FAQ documents (205) were considered to be low IDF terms. Earlier work in [9] has shown that stopword removal using a stopword list from various IR platforms like Terrier-3.5 can affect retrieval performance in SMS-Based FAQ retrieval. The normalisation parameter for BM25 was set to its default value of $b = 0.75$. For BM25F, the normalization parameter of each field was also set to 0.75 and these were $(b.0 = 0.75, b.1 = 0.75, b.2 = 0.75)$, representing the normalisation parameters for the $QUESTION$, $ANSWER$ and $FAQLog$ fields respectively. For PL2, the normalisation parameter was set to its default value of $c = 1$. For PL2F, the normalisation parameter for each field was set to $(c.0 = 1.0, c.1 = 1.0, c.2 = 1.0)$, representing the $QUESTION$, $ANSWER$ and $FAQLog$ fields respectively.

5.2 Experimental Investigation and Our Baseline Systems

In this study, we will investigate the following experiments:

EXV1: In this experiment, we are testing our proposed enrichment strategies. This was achieved by comparing the retrieval performance in terms of MRR and recall on the enriched collections of FAQ documents and a collection of non-enriched FAQ documents. We describe how the FAQ documents were enriched using the training set in the next section. A description of how we split the SMS query log into training and testing sets is also provided in the next section. To carry out this investigation, we used the retrieval settings described in Section 5.1. We built an index for each enriched collection of FAQ documents separately using the three fields ($QUESTION$, $ANSWER$ and $FAQLog$) so that we can use field-based weighting models such as BM25F [16] and PL2F [10] for retrieval (60 indices in total). As a baseline, we created two different indices of the original FAQ documents (non-enriched FAQ documents) using the two fields ($QUESTION$ and $ANSWER$). In the first index, we indexed the questions (Q) only and in the second index, we indexed both the question and answer (Q and A). For each index of the enriched FAQ documents, we used the associated testing set to make two runs using BM25F and PL2F as our weighing models. For each index of the non-enriched FAQ documents, we also used the 10 testing sets to make 2 runs using BM25F and PL2F as our weighting models. For this investigation, all the field weights parameters were intentionally set to 1 ($w.0 = 1, w.1 = 1, w.2 = 1$), where ($w.0, w.1$ and $w.2$) represent the

---

[2] http://terrier.org/

$QUESTION$, $ANSWER$ and $FAQLog$ field weights respectively. The field-based weighting models $BM25F$ and $PL2F$ are known to yield the same retrieval scores as their non field-based counterpart (BM25 and PL2 respectively) when all field weights are set to 1. To illustrate this, we also made two runs on the indexed collections with each testing set using BM25 and then PL2 as our weighting models.

EXV2: In this experiment, we investigate whether we can do better by optimising the field weights for the enriched FAQ documents collections. It is well known that significant gain in relevance can be obtained if the field weight parameters are properly optimised [15,16]. In our investigation, we use $EXV1$ as our baseline systems. We then optimise the field weights for all the enriched collections. A description of how the field weights were optimised can be found in the next section. We then perform retrieval on these enriched FAQ document collections using the associated testing set with the field weights for BM25F and PL25F set to their new optimal values.

EXV3: In experiments $EXV1$ and $EXV2$ we also investigated the effect of changing the size of the training set. In carrying out these experiments, three different collections that were enriched with queries of varying sizes were used for each testing set. A description of how these collections were created is detailed in the next section.

EXV4: To compare our approach with traditional approaches (e.g query expansion) normally used to resolve the term mismatch problem, we used the collection enrichment approach first introduced by Kwok et al. [8]. Collection enrichment is a form of query expansion where a high quality external collection is used to expand the original query terms and then retrieves from the local collection using the expanded query [8]. A local collection refers to the collection from which the final retrieved documents are retrieved. In the collection enrichment approach, we first performed retrieval on an external collection of HIV/AIDS documents crawled from the web. We crawled web pages that have a strong focus on HIV/AIDS frequently asked questions. Each web page crawled was indexed as a single document. In total, we had 3648 web page documents. For example, from *www.avert.org*, we were able to crawl 259 web documents. We provide examples of some of the domains and pages crawled in Table 3. In our collection enrichment approach, we used the Terrier Divergence From Randomness (DFR) Bo1 (Bose-Einstein 1) model to select the 10 most informative terms from the top 3 returned documents as expansion terms. These 10 new

Table 3. Examples of some of the web pages that were crawled from the web to use as an external collection for query expansion using collection enrichment approach

Web Page	Uniform Resource Locator (URL)
Avert : AVERTing HIV and AIDS	*http : //www.avert.org*
FAQ |AIDS Foundation of South Africa	*http : //www.aids.org.za*
What everyone should know about HIV	*http : //www.hivaware.org.uk*
$AIDS_\star$gov	*http : //www.aids.gov*

terms together with the original query terms were used for retrieval on the non enriched FAQ documents collection.

5.3   FAQ Documents Enrichment and Field Weights Optimisation

Our main contribution in this work as described in Section 1 is to demonstrate that using a field-based model to enrich the FAQ documents with additional terms from potential users of our FAQ retrieval system can alleviate the term mismatch problem that arises in our FAQ retrieval system. In order to achieve the above goals, we identified the following research hypotheses:

HP1: Enriching the FAQ documents with additional terms from queries for which the true relevant question-answer pair is known would increase the Mean Reciprocal Rank (MRR) and the overall recall in our FAQ retrieval system. Our intuition is that, additional terms introduced would help to reduce the term mismatch between the queries and the FAQ documents.

HP2: Increasing the number of queries used in enriching the FAQ documents would increase the (MRR) and the overall recall because additional terms introduced in the collection would help to alleviate the term mismatch problem.

To test hypotheses *HP1* and *HP2*, we produced 10 random splits of the 750 matched SMS queries into a training set of 600 queries and a test set of 150 queries. These SMS queries were first corrected for spelling errors, so that such a confounding variable does not influence the outcome of our experiments. We plan to incorporate a spelling correction approach to our system in the future.

To test HP2, we additionally split the 600 training queries into three sets of 200 and incrementally combined them to create training sets of size 200, 400 and 600 queries (hereafter referred to as 200SMSes, 400SMSes and 600SMSes). 400SMSes is therefore a superset of 200SMSes and 600SMSes is a superset of 400SMSes. This process was chosen as it emulates the temporal nature of query collection in a real system. For each train/test split, we created 6 (3 for term frequencies and the other 3 for term occurrences) enriched collections (corresponding to 200SMSes, 400SMSes and 600SMSes) using the two enrichment approaches described in Section. 3. In total, we created 60 different enriched FAQ documents collections.

In order to infer whether using field-based weighting models does indeed help in the overall retrieval performance in terms of MRR and recall, the weights for each field were optimised. Optimisation of these field weights is vital as significant gains in relevance can be obtained if the parameters are properly optimised [15,16]. We used the 10 random splits of the 600 SMS queries of training data for optimising the field weights. The test queries for each train/test split were naturally not used for optimisation of the field weights in order to avoid over-fitting. For each training set, we randomly selected 450 SMS queries and used these to enrich the FAQ documents using our two enrichment strategies proposed in Section 3, thus giving us 2 different enriched FAQ document collections for each training set. The remaining 150 SMS queries were left for optimising the field weights.

Table 4. The mean and standard deviation for the field weights $(w.1 = 1)$

Weighting Model	Enrichment Strategy	Mean Field Weights	Standard Deviation
PL2F	Term Occurrence	$w.0 = 6.68, w.2 = 5.74$	$stdv.0 = \pm3.18, stdv.2 = \pm2.53$
	Term Frequency	$w.0 = 5.53, w.2 = 7.04$	$stdv.0 = \pm3.33, stdv.2 = \pm2.97$
BM25F	Term Occurrence	$w.0 = 5.98, w.2 = 5.94$	$stdv.0 = \pm3.68, stdv.2 = \pm3.06$
	Term Frequency	$w.0 = 4.02, w.2 = 6.98$	$stdv.0 = \pm2.50, stdv.2 = \pm3.41$

(a) Question, Answer and FAQLog Fields (w.2=4).

(b) Question, Answer and FAQLog Fields (w.1=1).
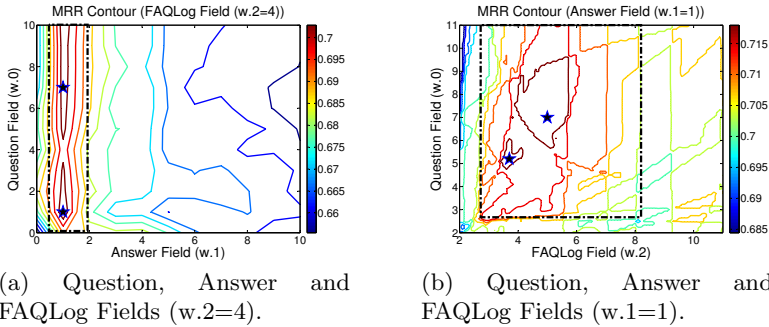
Fig. 1. The $\star$ denotes the region of highest MRR in relation to field weights $w.0, w.1$ and $w.2$ in this particular contour plots that were chosen randomly from our results. The higher MRR values for all the other random splits are inside the dotted rectangles.

In optimising the field weights, we used the Terrier-3.5 Information Retrieval (IR) platform. First we indexed the enriched collections separately without stopword removal and using the full Porter stemming algorithm. We then performed our optimisation using the Robust Line Search (RLS) strategy as described in [15]. For both BM25F and PL2F, we performed an initial scan of the field weights parameters $w.0, w.1$ and $w.2$ ($QUESTION, ANSWER$ and $FAQLog$ fields respectively) to determine the optimal values of these field weights with respect to a higher Mean Reciprocal Rank (MRR). In our initial scan, the field weights were varied linearly from 0.0 to 10.0 in steps of 1. Higher MRR values for the first scan were obtained when the $ANSWER$ field was set to 1 for most of the collections as shown in Figure 1.(a) (the $\star$ denotes the region of the highest MRR). For the $QUESTION$ and $FAQLog$ fields, higher MRR values were obtained when these fields were set to 2 or higher (Figure 1.(b)).

We then set a second starting point for each field weight to ($w.0 = 2.0, w.1 = 1.0, w.2 = 2.0$). Because the optimal value of the $ANSWER$ field was 1, this field was fixed while the others were varied linearly from 2.0 to 11.0 in steps of 0.1 for the second RLS. We increased the search space by varying parameters in steps of 0.1 instead of 1 so that we do not lose the global maximum. The above procedure was repeated for all the 10 random splits of training data. The optimal values of the field weights for these 10 random splits of training data were averaged to arrive at the final values of the field weights to use in testing our hypotheses *HP1* and *HP2*. Table 4 shows the mean and standard deviation of the field weights that we will use in our experimental investigation. It is worth pointing

out that these values were averaged taking into consideration that small changes in the parameter values of these models are known to produce small changes in the accuracy of relevance [15]. Our analysis of the various contour plots also show that the mean field weights in Table 4 are also within the region of higher MRR values that is bounded by the dotted rectangle in Figure 1.(b) for all the training samples.

6   Experimental Results and Evaluation

Table 5 summarises our experimental evaluation for research hypotheses *HP1* and *HP2*. As highlighted in [16], we can see that when setting the field weights to one (not optimised, $EXV1$), there in no improvement in retrieval performance in terms of MRR and recall for the field-based weighting models over the non field-based weighting models counterpart(BM25 and BM25F as well as PL2 and PL2F). Similar findings were also observed for the new enriched FAQ documents. However, there is a significant improvement in the retrieval performance (t-test, $p < 0.05$ for MRR) when the FAQ documents are enriched ($EXV1$).

There was a statistically significant (t-test, $p < 0.05$) increase in recall from around 0.2400 for non enriched FAQ documents to more than 0.4900 for the enriched FAQ documents. An increase in recall implies a reduction in term mismatch because previously un-retrieved documents have been retrieved. The benefit of using field-based weighting models is only realised after the field weights have been optimised ($EXV2$) as highlighted in Table 5. Higher recall values ranging from 0.68 to 0.77 and MRR values ranging from 0.67 to 0.73 were recorded, depending on the enrichment strategy. One plausible explanation for an increase in retrieval performance after optimising weights is that the fields of high importance (Question and FAQLog fields) have been assigned field weights of more than one, thus increasing the importance of term frequencies within those fields. As shown in Table 5, using the question field only without the answer field yielded better retrieval performance, suggesting that this field is more important than the answer field. Similar findings were also reported in [9].

Moreover, higher MRR values were obtained when enriching the FAQ documents using the query term frequencies rather than the query term occurrence (t-test p value for MRR ($p < 0.05$)). This is consistent with the above findings because the term frequencies approach just increases the term frequencies of repeating queries within the FAQLog field (similar to increasing the field weights). Finally, an increase in the size of the collection used to enrich the FAQ documents resulted in a slight increase in the average MRR (averaged across the 10 train/test partitions) for both PL2F and BM25F ($EXV3$). However, only the increase from 200 to 400 and 200 to 600 training SMS queries was statistically significant (t-test, $p < 0.05$), suggesting that adding more training SMS queries in the new field does indeed help to alleviate the term mismatch problem. Our approach performs better compared to query expansion ($EXV4$) using collection enrichment (t-test, $p < 0.05$) . This is because, the expansion terms were selected automatically without relevance judgement of the source documents.

Table 5. The mean retrieval performance for each Collection. Significant improvement in MRR and Recall if the FAQ documents are enriched with queries over non enriched FAQ documents, as denoted by $*$ (t-test, $p < 0.05$). Also, the was significant improvement in MRR and recall if field weights were optimised compared to non optimised field weights, as denoted by $**$ (t-test, $p < 0.05$).

Evaluation	Collection	Enrichment Strategy	Weighting Model	Field Weights ($w.1 = 1$)	Test Evaluation Measure		
					MRR	MAP	Recall
EXV1	Q(Only)	No Enrichment	BM25F/BM25	$w.0 = 1$	0.4312	0.2197	0.2495
	Q and A	No Enrichment	BM25F/BM25	$w.0 = 1$	0.4106	0.2302	0.2380
EXV4	Q(Only) and QE	Query Expansion	BM25F/BM25	$w.0 = 1$	0.4162	0.2022	0.2528
	Q ,A and QE	Query Expansion	BM25F/BM25	$w.0 = 1$	0.4317	0.2692	0.2974
EXV1 and EXV3	Q,A and 200SMS	Term Occurrence	BM25F/BM25	$w.0 = 1, w.2 = 1$	0.6120	0.4878	0.4951$*$
	Q,A and 400SMS				0.6614	0.4913	0.5466$*$
	Q,A and 600SMS				0.6608	0.5039	0.5924$*$
EXV2 and EXV3	Q,A and 200SMS	Term Occurrence	BM25F	$w.0 = 5.98, w.2 = 5.94$	**0.6774**	**0.5741**	**0.6772**$**$
	Q,A and 400SMS				**0.6692**	**0.5867**	**0.7089**$**$
	Q,A and 600SMS				**0.6666**	**0.5935**	**0.7009**$**$
EXV1 and EXV3	Q,A and 200SMS	Term Frequency	BM25F/BM25	$w.0 = 1, w.2 = 1$	0.6492	0.5146	0.5327$*$
	Q,A and 400SMS				0.6833	0.5491	0.5765$*$
	Q,A and 600SMS				0.6921	0.5435	0.6043$*$
EXV2 and EXV3	Q,A and 200SMS	Term Frequency	BM25F	$w.0 = 4.02, w.2 = 6.98$	**0.6847**	**0.6035**	**0.6902**$**$
	Q,A and 400SMS				**0.7179**	**0.6455**	**0.7546**$**$
	Q,A and 600SMS				**0.7315**	**0.6747**	**0.7484**$**$
EXV1	Q(Only)	No Enrichment	PL2F/PL2	$w.0 = 1$	0.4526	0.2720	0.2545
	Q and A	No Enrichment	PL2F/PL2	$w.0 = 1$	0.4106	0.2438	0.2711
EXV4	Q(Only) and QE	Query Expansion	PL2F/PL2	$w.0 = 1$	0.4297	0.2552	0.2815
	Q ,A and QE	Query Expansion	PL2F/PL2	$w.0 = 1$	0.4430	0.2627	0.2764
EXV1and EXV3	Q,A and 200SMS	Term Occurrence	PL2F/PL2	$w.0 = 1, w.2 = 1$	0.6068	0.5074	0.5841$*$
	Q,A and 400SMS				0.6310	0.5272	0.6168$*$
	Q,A and 600SMS				0.6831	0.5413	0.6340$*$
EXV2 and EXV3	Q,A and 200SMS	Term Occurrence	PL2F	$w.0 = 6.68, w.2 = 5.74$	**0.6766**	**0.5866**	**0.6950**$**$
	Q,A and 400SMS				**0.6938**	**0.6093**	**0.7188**$**$
	Q,A and 600SMS				**0.7004**	**0.6187**	**0.7465**$**$
EXV1 and EXV3	Q,A and 200SMS	Term Frequency	PL2F/PL2	$w.0 = 1, w.2 = 1$	0.6213	0.5432	0.5941$*$
	Q,A and 400SMS				0.6580	0.5535	0.6268$*$
	Q,A and 600SMS				0.6990	0.5848	0.6484$*$
EXV2 and EXV3	Q,A and 200SMS	Term Frequency	PL2F	$w.0 = 5.53, w.2 = 7.04$	**0.6701**	**0.6134**	**0.7246**$**$
	Q,A and 400SMS				**0.7112**	**0.6515**	**0.7585**$**$
	Q,A and 600SMS				**0.7254**	**0.6892**	**0.7713**$**$

This has some disadvantages as some queries might be expanded with irrelevant terms. Despite some of the disadvantages, a slight gain in MRR and recall was observed when the question and answer field were used and query expansion applied. However, there was a decrease in retrieval performance when only the question field was used, suggesting that the terms from the external collection might be adding noise to the original query.

7   Conclusions

In this paper we described a field-based approach to reduce the term mismatch problem in our SMS-Based FAQ retrieval system dealing with questions related to HIV and AIDS. Our experiments show that the inclusion of a field derived from logs of SMS queries for which the true relevant question-answer pair is known substantially improves the recall compared to query expansion using the collection enrichment approach. An increase in recall verified that the term mismatch did indeed significantly decrease (according to t-test) with the proposed approach.

In addition, we investigated how the number of queries used to enrich the FAQ documents affected performance. We saw a statistically significant increase in both recall and the average MRR when the number of queries used to enrich the FAQ documents were increased from 200 to 400 and 200 to 600. This results validates our second hypothesis $HP2$. An increase of training queries from 400 to 600 did not result in statistically significant improvement in MRR and recall. We plan to carry out further investigation with more queries to determine the point where there is no gain in retrieval performance even when the number of training queries is increased.

References

1. Billerbeck, B., Scholer, F., Williams, H.E., Zobel, J.: Query Expansion using Associated Queries. In: Proc. of CIKM (2003)
2. Billerbeck, B., Zobel, J.: Document Expansion Versus Query Expansion For Ad-hoc Retrieval. In: Proc. of ADCS (2005)
3. Fang, H.: A Re-examination of Query Expansion Using Lexical Resources. In: Proc. ACL:HLT (2008)
4. Hammond, K., Burke, R., Martin, C., Lytinen, S.: FAQ Finder: A Case-Based Approach to Knowledge Navigation. In: Proc. of CAIA (1995)
5. Jeon, J., Croft, W.B., Lee, J.H.: Finding Similar Questions in Large Question and Answer Archives. In: Proc. of CIKM (2005)
6. Kim, H., Lee, H., Seo, J.: A Reliable FAQ Retrieval System Using a Query Log Classification Technique Based on Latent Semantic Analysis. Info. Process. and Manage. 43(2), 420–430 (2007)
7. Kim, H., Seo, J.: High-Performance FAQ Retrieval Using an Automatic Clustering Method of Query Logs. Info. Process. and Manage. 42(3), 650–661 (2006)
8. Kwok, K.L., Chan, M.: Improving Two-Stage Ad-hoc Retrieval for Short Queries. In: Proc. of SIGIR (1998)
9. Leveling, J.: On the Effect of Stopword Removal for SMS-Based FAQ Retrieval. In: Bouma, G., Ittoo, A., Métais, E., Wortmann, H. (eds.) NLDB 2012. LNCS, vol. 7337, pp. 128–139. Springer, Heidelberg (2012)
10. Macdonald, C., Plachouras, V., He, B., Lioma, C., Ounis, I.: University of Glasgow at WebCLEF 2005: Experiments in Per-Field Normalisation and Language Specific Stemming. In: Proc. of CLEF (2006)
11. Moreo, A., Navarro, M., Castro, J.L., Zurita, J.M.: A High-Performance FAQ Retrieval Method Using Minimal Differentiator Expressions. Know. Based Syst. 36, 9–20 (2012)
12. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Lioma, C.: Terrier: A High Performance and Scalable Information Retrieval Platform. In: Proc. of OSIR at SIGIR (2006)
13. Plachouras, V., Ounis, I.: Multinomial Randomness Models for Retrieval with Document Fields. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECiR 2007. LNCS, vol. 4425, pp. 28–39. Springer, Heidelberg (2007)
14. Porter, M.F.: An Algorithm for Suffix Stripping. Elec. Lib. Info. Syst. 14(3), 130–137 (2008)
15. Robertson, S., Zaragoza, H.: The Probabilistic Relevance Framework: BM25 and Beyond. Found. Trends Info. Retr. 3(4), 333–389 (2009)

16. Robertson, S., Zaragoza, H., Taylor, M.: Simple BM25 Extension to Multiple Weighted Fields. In: Proc. of CIKM (2004)
17. Singhal, A., Pereira, F.: Document Expansion for Speech Retrieval. In: Proc. of SIGIR (1999)
18. Sneiders, E.: Automated FAQ Answering: Continued Experience with Shallow Language Understanding. Question Answering Systems. In: Proc. of AAAI Fall Symp. (1999)
19. Sneiders, E.: Automated FAQ Answering with Question-Specific Knowledge Representation for Web Self-Service. In: Proc. of HSI (2009)
20. Voorhees, E.M.: Query Expansion Using Lexical-Semantic Relations. In: Proc. of SIGIR, pp. 61–69 (1994)
21. Whitehead, S.D.: Auto-FAQ: an Experiment in Cyberspace Leveraging. Comp. Net. and ISDN Syst. 28(1-2), 137–146 (1995)
22. Xue, X., Jeon, J., Croft, W.B.: Retrieval Models for Question and Answer Archives. In: Proc. of SIGIR (2008)

Exploring Domain-Sensitive Features
for Extractive Summarization
in the Medical Domain

Dat Tien Nguyen[1] and Johannes Leveling[2]

[1] University of Engineering and Technology (UET)
Vietnam National University
Hanoi, Vietnam
datnt88@gmail.com
[2] Centre for Next Generation Localisation (CNGL)
School of Computing
Dublin City University
Dublin 9, Ireland
johannes.leveling@computing.dcu.ie

Abstract. This paper describes experiments to adapt document summarization to the medical domain. Our summarizer combines linguistic features corresponding to text fragments (typically sentences) and applies a machine learning approach to extract the most important text fragments from a document to form a summary. The generic features comprise features used in previous research on summarization. We propose to adapt the summarizer to the medical domain by adding domain-specific features. We explore two types of additional features: medical domain features and semantic features. The evaluation of the summarizer is based on medical articles and targets different aspects: i) the classification of text fragments into ones which are important and ones which are unimportant for a summary; ii) analyzing the effect of each feature on the performance; and iii) system improvement over our baseline summarizer when adding features for domain adaptation. Evaluation metrics include accuracy for training the sentence extraction and the ROUGE measure computed for reference summaries. We achieve an accuracy of 84.16% on medical balanced training data by using an IB1 classifier. Training on unbalanced data achieves higher accuracy than training on balanced data. Domain adaptation using all domain-specific features outperforms the baseline summarization wrt. ROUGE scores, which shows the successful domain adaptation with simple means.

Keywords: Automatic Summarization, Sentence Extraction, Machine Learning, ROUGE.

1 Introduction

As the problem of information overload still grows and the amount of information available on the internet increases, text summarization is becoming a more important Natural Language Processing (NLP) task. The objective of summarizing a text document automatically is to provide a shorter version of a document,

which typically has 10-30% of the original text length [1] and still contains the most important information in a coherent form. In contrast, manual summarization is costly and time-consuming.

In spite of extensive research on automatic text summarization, there is still limited research on domain-specific summarization and on domain-adaptation of summarization for domains such as the medical, chemical, or legal domain. This paper describes experiments for adapting a generic summarizer to document summarization in the medical domain. The summarizer produces an extractive summary consisting of the most important sentences in the original document. We employ various features that have been described in previous research for generic summarizers to form a baseline summarizer. The main contribution of this paper is the investigation of simple domain-specific and semantic features for adapting summarization to the medical domain.

For domain adaptation, we extend the generic summarizer with domain-specific features. We investigate whether medical or semantic features or their combination can contribute to improve the produced summaries in the medical domain and automatically evaluate the quality of the summaries. We perform summarization experiments on documents from the medical domain and analyze the results in detail, by measuring the impact of each individual domain-specific feature. Then we conduct experiments using a combination of all features and measure the improvement of system accuracy when using additional features. Finally, we measure the ROUGE-N score to automatically evaluate the summarization quality with reference to the reference summary from the document. We obtain 84.08% accuracy on balanced training data.

The rest of this paper is organized as follows: Related work is briefly reviewed in Section 2. In Section 3 we introduce our text summarizer. The evaluation approach is presented in Section 4, followed by an analysis and discussion of results in Section 5. The paper concludes with Section 6.

2   Related Work

Automated summarization by IR approaches as well as machine learning approaches are well established research topics since the 1950s. Most traditional approaches [2,3,4], view text summarizing as a text extraction task, where portions of a document are combined into a summary.

Several surveys about automatic text summarization [1,5,6] describe traditional and modern approaches for text summarization. One common approach is to derive features from text fragments as input for machine learning to a train a sentence classifier. A strong baseline system which shared some generic features with our proposed summarizer has been analyzed by Nenkova [7].

We briefly describe some recent systems that related to our work. Conroy and O'Leary [8] show how a Hidden Markov Model (HMM) can be used for automatic summarization. They experimented with generic features such as the skimming feature (the position of sentence in the text), the number or terms in a sentence and the probability of a term estimated from the input that described in earlier work. However, they built training data based on manually judged summaries produced by a single person.

Lin et al. [9] used SUMMARIST [1] to compare the effects of eighteen difference features on summarization. These features are also included in our summarizer: proper names, date/time terms, pronouns, prepositions and quotes. They used the machine learning algorithm C4.5 to evaluate single features as well as an optimal feature combination. The performance of the individual methods and combination showed that the best scoring result with respect to F-score is the feature combination.

There have been many automatic summarization techniques introduced since the first research on using machine learning techniques [10]. However the main idea of later work concentrates essentially on the comparison of different learning algorithms and the way how to categorize feature classes [6].

Evaluation of summarization is notoriously difficult. Summarization tasks such as DUC[1] or the Text Analysis Conference (TAC) encouraged research by providing large documents corpora and summaries. However, judgments for evaluation typically relied on human annotators or assessors. The INEX[2] task for evaluating retrieval of snippets (short extracts from documents), introduced in 2011, is also related to summarization. This task aims at evaluating the snippet extraction to investigate if a user can understand the content of a document without reading the full document.

There are several summarization systems that have dealt with summarization of documents in the medical domain. Yang et al. [11] built and evaluated a query-based automatic summarizer on the domain of mouse genes studied in micro-array experiments. Their system implemented sentence extraction following the approach proposed by Edmundson [3]. However, before ranking sentences by aggregating features such as special keywords, sentence length, and cue phrases, the gene set was clustered into groups based on free text, and MeSH and GO terms belonging to a gene ontology. They used Medline abstracts to investigate ranked sentences of the summary output.

The Technical Article Summarizer (TAS) [12] automatically generates a summary that is suitable for patient characteristics when the input to the system is a patient record and journal articles. This helps physicians or medical experts to easily find information relevant to the patient's situation.

Our REZIME Summarizer system utilizes machine learning to automatically determine the importance of a sentence based on features. We implemented a large set of sentences features that have been described in previous research, but we use a different approach to incorporate these features in a training model. In this paper, we are particularly interested in how these established and proven, but generic features can be extended for domain-adaptation. We think that domain-specific knowledge and semantic information will help to adapt the summarization to the domain we chose for our experiments, the medical domain. We employed a subset of a collection of documents from BioMed Central (BMC)[3], which contain the (reference) abstracts and the full body of texts.

---

[1] http://www.nist.gov/tac/
[2] https://inex.mmci.uni-saarland.de/
[3] http://www.biomedcentral.com/info/about/datamining/

3   REZIME Summarizer

The objective of the REZIME summarizer[4] is to select the most important sentences as a summary that represents the original text.

3.1   System Architecture

The general architecture of our automatic summarizer is shown in Figure 1. A given text input is tokenized to split up the text into different linguistic units (e.g. paragraphs, sentences, phrase, and words). Thus, the *tokenization* includes tasks such as paragraph detection (e.g. based on text markup such as $<p>$), sentence boundary detection, and phrase recognition (e.g. based on frequently occurring word n-grams). The next processing step is feature calculation, where for each sentence, an aggregated score is computed which is in turn based on scores assigned to words (e.g. for capitalization), phrases (e.g. occurrence in a specific word list or lexicon), or the sentence itself (e.g. sentence length). The resulting score is then normalized into [0, ..., 1].

To create training data, the features for all sentences in the document's body of text are computed. The corresponding class is obtained by comparing the document sentence to all sentences in the reference abstract and computing their similarity. If the similarity (e.g. a normalized term overlap score) exceeds a given threshold t (e.g. $t > 0.8$), the sentence is determined to be important for the summary (i.e. class 1); otherwise it is discarded (i.e. class 0).

The classifier is trained on a subset of the document collection and is then applied to the test documents. We experimented with different machine learning approaches provided by the Weka toolkit[5]. The summary is generated by presenting the N most important sentences in order of appearance in the document.
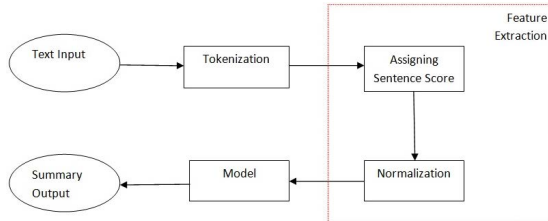
Fig. 1. REZIME system architecture

3.2   Feature Description

We investigate different sentence features that have proved useful in various automatic text summarization systems. To easily describe features used in REZIME, we divide all features into 2 groups: Term Checking Features and Non Term Checking features (i.e. all other features).

---

[4] REZIME is the Haitian Creole word for *summarize*.
[5] http://www.cs.waikato.ac.nz/ml/weka/

Term Checking Features: Term checking features are based on a check whether a text fragment (e.g. a sentence or paragraph) contains words or phrases with certain characteristics. The characteristics can relate to word occurrence in a list, capitalization information etc. For example, the title of an article often reveals the major subject of that document. Sentences containing terms from the title are likely to be good summarization candidates. A typical feature belonging to this group is the *title term feature*. For our generic baseline summarizer, we used many features that have been proposed in previous research described in some surveys of extractive text summarization techniques [13].

- *Basic Words Feature*: models word complexity and may be useful to produce summaries for non-expert users that are not overly familiar with more technical language [13]. The basic words list contains 1,040 common and frequently used words of every-day English.
- *Cue Phrase Feature*: checks for specified indicator phrases such as "importantly" or "in summary" in a sentence. These phrases may indicate that the sentence is good for summarization (positive phrases), or that it is not (negative phrases) [13].
- *Date/Time Feature*: checks for temporal expressions (e.g. weekdays or month names) in a sentence [9].
- *Named Entity Feature*: calculates the number of named entities (NE) that occur in each sentence. As NE taggers are quite slow, a more naive approach is taken here. Any word (except the first in a sentence), that starts with a capital letter is assumed to be a NE [9]. For English, proper nouns are capitalized and common nouns are not.
- *Preposition Feature*: tests if prepositions occur in a sentence [9], using a predefined list of prepositions from the part-of-speech tagged Brown corpus.
- *Pronoun Feature*: uses a predefined list to identify pronouns. Pronouns should be avoided in a summary unless the entities they referred to are also included or they are expanded into corresponding nouns [9].
- *Punctuation Feature*: calculates the proportion of punctuation tokens in all tokens in the sentence. If the value exceeds a specified threshold, the sentence is presumed to contain noise and will be unimportant for a summary.

The remaining question for a summarizer is how to transform these features into numeric values for a machine learning approach. This aspect has received little attention in research and most summarizers use a single transformation function to achieve this. For each of the term checking features described above, we can derive three feature variants to compute a score for a sentence s containing n_w occurrences of a word w, corresponding to a binary feature, the raw number of positive term checks, and the percentage of positive term checks:[6] i) occurrence ($score_s = 1$ if $w \in s$; 0 otherwise), ii) count ($score_s = n_w$), and iii) ratio ($score_s = n_w/|s|$).

---

[6] For simplicity, we refer to sentences and words, but the description can be generalized to include text fragments such as phrases.

Non Term Checking Features: The Non Term Checking Feature group includes more complex features compared to the above group. For a more detailed description of these features, the reader is again referred to the original literature.

- *Cluster Keyword Feature*: considers two significant words as related if they are separated by not more than five insignificant words. Important sentences will have large clusters of significant words; proposed by Luhn [2].
- *The Global Bushy Feature*: generates inter-document links based on similarity of paragraphs; paragraphs with many links share vocabulary with many other paragraphs and are important; proposed by Salton [14].
- *Number of Terms/Sentence Length Feature*: The number of terms in a sentence, assuming that too long or too short sentences are unimportant for a summary [3].
- *Skimming Feature*: The position of a sentence in a paragraph. The underlying assumption is that sentences occurring early in a paragraph are more important for a summary [13,15].
- *TS-ISF Feature*: Similar to TF-IDF, but works on the sentence level. Every sentence is treated like a document. Sentences which have a lot of keywords is likely included in summary [16,13].

Domain-adaptation Features: In addition to the features used for our generic summarizer (i.e. not adapted to a particular domain), we propose special features that improve summarization performance in the medical domain.

Medical Domain Features. Adaptation to the medical domain is often achieved by using formal ontologies or taxonomies such as MeSH (Medical Subject Headings). However, in the experiments described in this paper we focus on domain adaptation with relatively simple means.

- *Affix Presence Feature*: tests if a word contains affixes (i.e. word prefixes, infixes, and suffixes) from a manually created list of medical affixes for words derived from Latin and Greek. The idea for this feature is that medical technical terms are often derived from Latin or Greek and these terms would indicate the importance of a sentence. The affix list was manually compiled from medical literature and from the corresponding Wikipedia articles on medical terms. There are 897 affixes in this list. For example, the prefix *gastro-* means stomach (e.g. in *gastroenteritis*) and the suffix *-itis* means an inflammation (e.g. in *appendicitis*).
- *Domain Term Feature*: detects terms on a list of technical terms. The list of terms was collected from `http://www.medterms.com/` and comprises 16,477 medical terms. For example, the list contains the terms *gallbladder* (a small organ that aids mainly in fat digestion and concentrates bile produced by the liver) and *Zollinger-Ellison syndrome* (a disease caused by a non beta islet cell).

Semantic Features. The lexical database WordNet[7] has a high coverage of terms from medicine and biology (e.g. names of diseases or drugs). Thus, we define

---

[7] `http://wordnet.princeton.edu/`

semantic features based on WordNet, using Word Sense Disambiguation as proposed by [17].

- *Ambiguity Feature*: computes the sum of the number of WordNet senses for each term in a sentence, indicating highly ambiguous sentences.
- *Title Term Synonyms Feature*: uses the Adapted Lesk Algorithm [18] with the Lin measure [19] to find the best sense for each sense-disambiguated word in the document title. The list of synonyms is then added to the title terms list. This allows to compute a variant of the *Title Term Feature*.
- *Term Frequency Hierarchy Feature*: uses the Adapted Lesk algorithm to find the best sense for each term, based on the most frequent 100 terms in a document.[8] We then compute the depth of each noun and verb in the sentence based on the WordNet synset hierarchy. This approach is similar to Plaza's approach [20] applied to automatic summarization of news using WordNet concept graphs. However, we just compute the depth of each concept in WordNet and do not represent the document sentences as a graph. Moreover, concepts closer to the top level in the hierarchy correspond to a more generic meaning, which means this concept is likely to be included in summary.

We employed four formulas to compute numeric feature variants: i) minimum ($score = \min\{D(synset_i)\}$), ii) maximum ($score = \max\{D(synset_i)\}$), iii) average ($score = \Sigma_i D(synset_i)/HighFrequent(|s|)$), and iv) ratio ($score = \Sigma_i D(synset_i)/|s|$); where $i = 1, ..., 100$; $|s|$ is the number of terms in the sentence s, $D(synset_i)$ is the depth of the synset for word i in WordNet, and $HighFrequent(|s|)$ is the number of highly frequency terms in s. We refrain from experimenting with purely medical ontologies such UMLS, because of usage and access limitations (e.g. required license or high latency when using a web API).

4 Experiments

In this section, we describe the experiments using different machine learning algorithms to classify sentences and the final evaluation of the summarizer.

4.1 Training Data

We employed a subset of the BMC document collection[9] for our experiments. For each document article, we consider the abstract as a reference summary to be used in training and evaluating the summarizer. We view sentence extraction as a classification problem with two classes: 0 (discard the sentence for the summary) and 1 (keep the sentence). To generate the classes for the training data, we compute the normalized term overlap scores (T_{ovl}) between sentences $s1$ and $s2$ in the document and each sentence in the abstract, i.e. the number of shared terms between the sentences, normalized by the maximum length (see Equation 1). If a term overlap score exceeds the threshold of 0.8, the sentence will be included in the summary of the document and this sentence is classified belonging to class 1; the remaining sentences are class 0.

---

[8] Initial experiments were based on using all terms, but this proved to be inefficient.
[9] http://www.biomedcentral.com/info/about/datamining/

$$T_{ovl}(s1, s2) = \frac{|s1 \cap s2|}{max(|s1|, |s2|)} \tag{1}$$

An initial filtering step aims at generating training data of high quality. Documents which do not contain at least one sentence in class 1 were discarded. Therefore, we randomly selected 2,000 medical domain articles (101 Megabyte). After filtering, 1,263 documents remain as training data.

The number of sentences with a low term overlap score is always much larger than higher term overlap score sentence. This forms our first training set with unbalanced data. We also generated a balanced training set for training the classifier, i.e. where the number of class 1 instances is closer to or equal to the number of class 0 instances.

4.2   Evaluation and Results

Experiment 1: Balanced Training. We use the Weka[10] machine learning package to train on balanced training data. Weka supports a variety of machine learning algorithms (see, for example, [21]). We chose four machine learning algorithms to determine a strong baseline for the generic summarizer: IB1, Naive Bayes, Logistic Regression and Bayes Decision Tree. The balanced dataset, derived from processing 1,263 documents, has 15,108 instances, in which class 1 has 7,554 instances, the remainder belong to class 0. The accuracy of the classifier using 10-fold cross-validation is shown in Table 1.

Table 1. Training results on balanced and unbalanced data for IB1, Naive Bayes (NB), Logistic Regression (LR), and Bayes Decision Tree (BDT)

Training Data Instances (1/0)	Accuracy [%]			
	IB1	NB	LR	BDT
Balanced 15,108 (7,554/7,554)	**84.16**	69.08	76.05	**82.71**
Unbalanced 1 22,662 (7,554/15,108)	84.39	72.34	77.96	80.73
Unbalanced 2 37,770 (7,554/30,216)	87.18	76.65	83.31	84.31
Unbalanced 3 52,878 (7,554/45,324)	89.12	78.58	86.80	85.18

Experiment 2: Unbalanced Training. We conduct an experiment on unbalanced training data, where the the number of instances in class 0 varies. The results for each unbalanced training dataset are also shown in Table 1. As a result of experiments 1 and 2, we choose IB1 for the rest of our experiments, as it shows consistent and good performance over different data sets.

Experiment 3: Domain Adaptation. We investigate how good our proposed features are for adapting summarization to the medical domain compared to our generic (unadapted) baseline system, using the balanced training dataset and the IB1 classification. Both medical domain features and semantic features effectively improve the accuracy of the system, as shown in Table 2.

---

[10] http://www.cs.waikato.ac.nz/ml/weka/

Table 2. Accuracy for different additional features on the training data

Features	Accuracy [%]
Baseline	76.94
Baseline + ambiguity feature	79.34 (+2.40%)
Baseline + title term synonyms feature	77.99 (+1.05%)
Baseline + term frequency hierarchy feature	80.40 (+3.46%)
Baseline + all semantic features	83.09 (+6.15%)
Baseline + affix presence feature	78.20 (+1.26%)
Baseline + domain term feature	78.69 (+1.75%)
Baseline + all medical-domain features	78.77 (+1.83%)
Baseline + all other features	84.16 (+7.22%)

Experiment 4: ROUGE-N Evaluation. Lin introduced a set of metrics called Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [22], which has become a standard metric for automatic summarization evaluation.

In this evaluation experiment, we compute the correlation between the ROUGE scores for summaries with respect to the reference abstract in the full original document. This evaluation is based on a subset of 200 randomly selected documents (different from the training set), using the IB-1 classifier. We compare training on the balanced data with training on unbalanced datasets, varying the ratio of class 1 instances to class 0 instances (i.e. the ratio is in {1, 2, 4, 6, ..., 16}) to investigate the improvement of REZIME compared to the baseline.
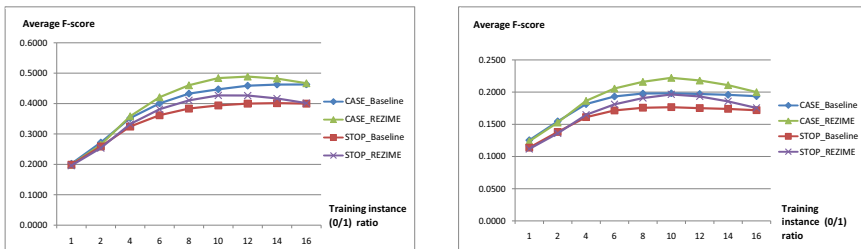
Fig. 2. Average F-Score for ROUGE 1 (left) and 2 (right)

In accordance to the conclusion of Lin 2004 [22], ROUGE-N (N = 2, 3) is useful for evaluation a text summarization system. For simplicity, we compute the ROUGE scores for preprocessed summaries using case folding to lowercase (CASE) and stopword removal (STOP). Figure 2 shows the F-score correlation analysis results on the 200 test documents. The x-axis in these figures shows the ratio of instances in class 1 versus instances in class 0 and the y-axis shows the average F-score.

We also report the compression ratio of generated summaries for each setting in Figure 3. This figure shows that with increasingly more unbalanced data, the length of the produced summary decreases, i.e. balanced training data would produce a more even distribution of sentences in class 0 and class 1 (almost 0.5), which leads to a longer summary.

5 Analysis and Discussion

Our evaluation results show that surprisingly, training on unbalanced data yields a higher accuracy compared to training on balanced data. This can at least in part be explained by the fact that with larger training data, more instances in the dominating class are classified correctly. The automatic evaluation approach shows that a considerably good accuracy can be achieved compared to evaluations based on costly manual judgments.

The additional features for domain adaptation improve summarization performance (accuracy) on the training data. Interestingly, the semantic features improve the performance more than the medical term features (+6.15% vs. +1.83%). The extension REZIME's feature set with both the semantic and the medical term features shows the best performance.

In our ROUGE evaluation, when training with balanced or unbalanced training data with a low ratio of class 0 and class 1 instances, the F-score is very low because the generated summary is longer than the abstract of document. In contrast, increasing the percentage of non-relevant sentences makes the generated summary to be shorter and more accurate. This leads to improvement of the results compared to the baseline system as well as the F-scores. The ROUGE performance on CASE normalized data is better than that using STOP (stopword removal). However, the F-score of the REZIME system is also higher than the baseline, when the training instances ratio increases. The ratio where instance 0 equals 10 fold instance 1 shows the best improvement. In summary, setting the ratio may control the compression rate of the produced output summaries and the summarization quality.
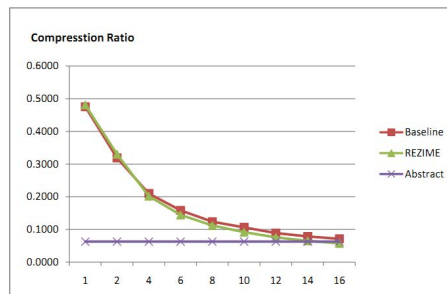
Fig. 3. Compression ratio in summarization output

6 Conclusions and Future Work

In this paper, we investigated extractive summarization and explored features for adapting generic summarization system to the medical domain. We implemented 19 features that are based on features used in generic automatic summarization systems. For each feature, feature variants were derived (e.g. the raw count vs. the ratio). We investigated the addition and combination of medical domain features and semantic features for domain adaptation to the medical domain.

In the experimental section, we trained the summarizer on different training sets, including training on balanced data and training on unbalanced data. We achieve an accuracy of 84.16% on medical balanced training data by using an IB1 classifier. Training on unbalanced data is considerably more accurate than training on balanced data (+4.96%). Adding the domain adaptation features increases accuracy by +7.22%, which illustrates that domain adaptation to the medical domain can be achieved with relatively simple means. To the best of our knowledge, the features we proposed for medical domain adaptation have not been investigated in detail for summarization, yet.

Finally, we conducted ROUGE-N evaluations ($N = 1, 2$) on 200 random medical documents. This evaluation showed that the improvement in accuracy observed in training the summarizer depends on the setting of training instances in the dataset and does not fully carry over to the automatic ROUGE-N evaluation of the summaries. However, overall, the F-score of the REZIME system (baseline features and all other features) outperforms the baseline system.

As part of our future work, we will investigate coreference resolution and replacing pronouns with the corresponding proper nouns to make the summaries more comprehensible and coherent. Initial experiments extending the classification problem to control the compression rate used n classes instead of two classes and showed a significant drop in accuracy. Thus, we plan to explore ranking sentences by the confidence factor of the classification. Finally, we want to research how larger taxonomies and ontologies such as MeSH or UMLS can contribute to improving summarization output.

References

1. Hovy, E., Lin, C.Y.: Automated text summarization in SUMMARIST. In: Mani, I., Maybury, M.T. (eds.) Advances in Automatic Text Summarization. MIT Press (1999)
2. Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of Research and Development 2(2), 159–165 (1958)
3. Edmundson, H.P.: New methods in automatic extracting. Journal of the ACM 16(2), 264–285 (1969)

4. Paice, C.D.: The automatic generation of literature abstracts: An approach based on the identification of self-indicating phrases. In: SIGIR 1981, pp. 172–191 (1981)
5. Nenkova, A., McKeown, K.: Foundations and trends in information retrieval. Automatic Summarization 5, 103–233 (2011)
6. Das, D., Martins, A.F.: A survey on automatic text summarization. Technical report, Literature Survey for the Language and Statistics II course at Carnegie Mellon University (2007)
7. Nenkova, A.: Automatic text summarization of newswire: lessons learned from the document understanding conference. In: AAAI 2005, pp. 1436–1441. AAAI Press (2005)
8. Conroy, J.M., O'Leary, D.P.: Text summarization via Hidden Markov Models. In: SIGIR 2001, pp. 406–407 (2001)
9. Lin, C.Y.: Training a selection function for extraction. In: Proceedings of the Eighth International Conference on Information and Knowledge Management, CIKM 1999, pp. 55–62. ACM, New York (1999)
10. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 68–73 (1995)
11. McKeown, K., Chang, S.F., Cimino, J., Feiner, S., Friedman, C., Gravano, L., Hatzivassiloglou, V., Johnson, S., Jordan, D., Klavans, J., Kushniruk, A., Patel, V., Teufel, S.: PERSIVAL, a system for personalized search and summarization over multimedia healthcare information. In: ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 331–340 (2001)
12. Yang, J., Cohen, A., Hersh, W.: Automatic summarization of mouse gene information by clustering and sentence extraction from MEDLINE abstracts. In: AMIA Annual Symposium, pp. 831–835 (2007)
13. Gupta, V., Lehal, G.: A survey of text summarization extractive techniques. Journal of Emerging Technologies in Web Intelligence 2(3) (2010)
14. Salton, G., Singhal, A., Mitra, M., Buckley, C.: Automatic text structuring and summarization. Inf. Process. Manage. 33(2), 193–207 (1997)
15. Brandow, R., Mitze, K., Rau, L.F.: Automatic condensation of electronic publications by sentence selection. Inf. Process. Manage. 31(5), 675–685 (1995)
16. Litvak, M., Last, M.: Graph-based keyword extraction for single-document summarization. In: Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization. MMIES 2008, pp. 17–24. Association for Computational Linguistics, Stroudsburg (2008)
17. Patwardhan, S., Banerjee, S., Pedersen, T.: Using measures of semantic relatedness for word sense disambiguation. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, pp. 241–257. Springer, Heidelberg (2003)
18. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordNet. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 136–145. Springer, Heidelberg (2002)
19. Lin, D.: An Information-Theoretic Definition of Similarity. In: Shavlik, J.W., Shavlik, J.W. (eds.) ICML, pp. 296–304. Morgan Kaufmann (1998)
20. Plaza, L., Díaz, A., Gervás, P.: Automatic summarization of news using Wordnet concept graphs. In: Proceedings of the IADIS International Conference Informatics, pp. 19–26 (2009)
21. Fattah, M.A., Ren, F.: Ga, mr, ffnn, pnn and gmm based models for automatic text summarization. Computer Speech & Language 23(1), 126–144 (2009)
22. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proc. ACL Workshop on Text Summarization Branches Out, p. 10 (2004)

A Corpus-Based Approach for the Induction of Ontology Lexica

Sebastian Walter, Christina Unger, and Philipp Cimiano

Semantic Computing Group, CITEC, Bielefeld University

Abstract. While there are many large knowledge bases (e.g. Freebase, Yago, DBpedia) as well as linked data sets available on the web, they typically lack lexical information stating how the properties and classes are realized lexically. If at all, typically only one label is attached to these properties, thus lacking any deeper syntactic information, e.g. about syntactic arguments and how these map to the semantic arguments of the property as well as about possible lexical variants or paraphrases. While there are lexicon models such as *lemon* allowing to define a lexicon for a given ontology, the cost involved in creating and maintaining such lexica is substantial, requiring a high manual effort. Towards lowering this effort, in this paper we present a semi-automatic approach that exploits a corpus to find occurrences in which a given property is expressed, and generalizing over these occurrences by extracting dependency paths that can be used as a basis to create *lemon* lexicon entries. We evaluate the resulting automatically generated lexica with respect to DBpedia as dataset and Wikipedia as corresponding corpus, both in an automatic mode, by comparing to a manually created lexicon, and in a semi-automatic mode in which a lexicon engineer inspected the results of the corpus-based approach, adding them to the existing lexicon if appropriate.

Keywords: ontology lexicalization, corpus-based approach, lemon.

1 Introduction

The structured knowledge available on the web is increasing. The Linked Data Cloud, consisting of a large amount of interlinked RDF datasets, has been growing steadily in recent years, now comprising more than 30 billion RDF triples[1]. Popular and huge knowledge bases exploited for various purposes are Freebase, DBpedia, and Yago.[2] Search engines such as Google are by now also collecting and exploiting structured data, e.g. in the form of knowledge graphs that are used to enhance search results.[3] As the amount of structured knowledge available keeps growing, intuitive and effective paradigms for accessing and querying

---

[1] `http://www4.wiwiss.fu-berlin.de/lodcloud/state/`
[2] `http://www.freebase.com/`, `http://dbpedia.org/`,
   `http://www.mpi-inf.mpg.de/yago-naga/yago/`
[3] `http://www.google.com/insidesearch/features/search/knowledge.html`

this knowledge become more and more important. An appealing way of accessing this growing body of knowledge is through natural language. In fact, in recent years several researchers have developed question answering systems that provide access to the knowledge in the Linked Open Data Cloud (e.g. [8], [13], [14], [2]). Further, there have been some approaches to applying natural language generation techniques to RDF in order to verbalize knowledge contained in RDF datasets (e.g. [10], [12], [4]). For all such systems, knowledge about how properties, classes and individuals are verbalized in natural language is required. The *lemon* model[4] [9] has been developed for the purpose of creating a standard format for publishing such lexica as RDF data. However, the creation of lexica for large ontologies and knowledge bases such as the ones mentioned above involves a high manual effort. Towards reducing the costs involved in building such lexica, we propose a corpus-based approach for the induction of lexica for a given ontology which is capable of automatically inducing an ontology lexicon given a knowledge base or ontology and an appropriate (domain) corpus. Our approach is supposed to be deployed in a semi-automatic fashion by proposing a set of lexical entries for each property and class, which are to be validated by a lexicon engineer, e.g. using a web interface such as *lemon source*[5].

As an example, consider the property `dbpedia:spouse` as defined in DB-pedia. In order to be able to answer natural language questions such as *Who is Barack Obama married to?* we need to know the different lexicalizations of this property, such as *to be married to*, *to be the wife of*, and so on. Our approach is able to find such lexicalizations on the basis of a sufficiently large corpus. The approach relies on the fact that many existing knowledge bases are populated with instances, i.e. by triples relating entities through properties such as the property `dbpedia:spouse`. Our approach relies on such triples, e.g. ⟨`dbpedia:Barack_Obama`, `dbpedia:spouse`, `dbpedia:Michelle_Obama`⟩ to find occurrences in a corpus where both entities, the subject and the object, are mentioned in one sentence. On the basis of these occurrences, we use a dependency parser to parse the relevant context and generate a set of lexicalized patterns that very likely express the property or class in question.

The paper is structured as follows: in Section 2 we present the general approach, distinguishing the case of inducing lexical entries for properties and for classes. The evaluation of our approach with respect to 80 pseudo-randomly selected classes and properties is presented in Section 3. Before concluding, we discuss some related work in Section 4.

2   Approach

Our approach[6] is summarized in Figure 1. The input is an ontology and the output is a lexicon in *lemon* format for the input ontology. In addition, it relies on an RDF knowledge base as well as a (domain) corpus.

---

[4] For detailed information, see `http://lemon-model.net/`

[5] `http://monnetproject.deri.ie/lemonsource/`

[6] Available at `https://github.com/swalter2/knowledgeLexicalisation`
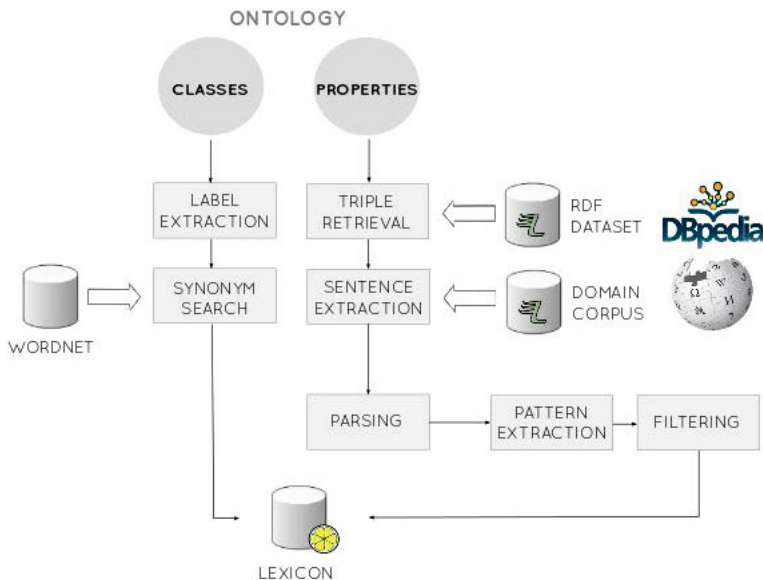
Fig. 1. System overview

The processing differs for properties and classes. In what follows, we describe the processing of properties, while the processing of classes, which does not rely on the corpus, is explained below in Section 2.5. For each property to be lexicalized, all triples from the knowledge base containing this property are retrieved. The labels of the subject and object entities of these triples are then used for searching the corpus for sentences in which both occur. Based on a dependency parse of these sentences, patterns are extracted that serve as basis for the construction of lexical entries. In the following, we describe each of the steps in more detail.

2.1   Triple Retrieval

Given a property, the first step consists in extracting from the RDF knowledge base all triples containing that property. In the case of DBpedia, for the property `dbpedia:spouse`, for example, 44 197 triples are found, including the following[7]:

- (`resource:Barack_Obama`, *dbpedia:spouse*, `resource:Michelle_Obama`)
- (`resource:Alexandra_of_Denmark`, *dbpedia:spouse*, `resource:Edward_VII`)
- (`resource:Hilda_Gadea`, *dbpedia:spouse*, `resource:Che_Guevara`)
- (`resource:Mel_Ferrer`, *dbpedia:spouse*, `resource:Audrey_Hepburn`)

---

[7] Throughout the paper we use the prefixes `dbpedia` and `resource` for `http://dbpedia.org/ontology/` and `http://dbpedia.org/resource/`, respectively.

2.2   Sentence Extraction and Parsing

For each triple (s, p, o), that was extracted for a property p, we retrieve all sentences from the domain corpus in which the labels of both entities s and o occur. This step is performed relying on an inverted index. An example sentence extracted from Wikipedia for the subject/object pair *Barack Obama* and *Michelle Obama* is the following:

> *The current First Lady is Michelle Obama, wife of Barack Obama.*

Each of the retrieved sentences is parsed with the pre-trained Malt dependency parser[8]. In order to avoid errors in parsing, the entity occurrences are replaced with a single word. For example, *Queen Silvia of Sweden* is replaced with *QueenSilviaofSweden*; this ensures that it is tagged as a named entity. Once a sentence has been parsed, the dependency parse is added to the index. This speeds up the process when the same sentence is retrieved again later.

From the dependency parses, we extract all paths that connect the entities in question. For the sentence above, for example, the following path connecting *Barack Obama* and *Michelle Obama* is found:

The current First Lady is MichelleObama , wife of BarackObama .
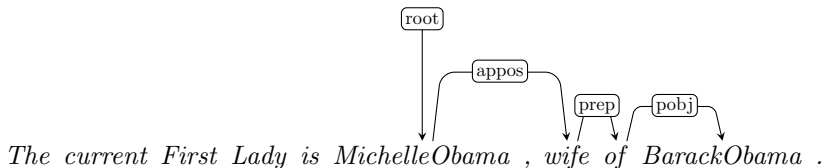
2.3   Pattern Generation, Postprocessing and Filtering

On the basis of the discovered dependency paths, patterns are generated by abstracting over the specific entities occuring in the parse. The above mentioned path would for instance be generalized to:
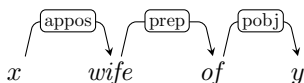
In addition, the generalized patterns are postprocessed, e.g. by removing determiners such as *the*. To avoid unnecessary noise, only patterns with a length of at least three but not longer than six tokens are accepted. Also, if the entities x or y are related to another token by nn, i.e. are modifiers, the pattern is not considered. Additional processing, such as subsuming similar patterns under a single one, are planned but not yet implemented.

Finally, for each property we compute the relative frequency of the found patterns, i.e. the number of sentences that yielded a certain pattern in relation to the overall number of sentences for that property. We then consider only those patterns that occur at least twice and surpass a certain threshold θ, which is determined empirically in Section 3.3 below.

---

[8] http://www.maltparser.org/

2.4   Generation of Lexical Entries

All patterns found by the above process, whose relative frequency is above a given threshold θ, are then transformed into a lexical entry in *lemon* format. For instance, the above mentioned pattern is stored as the following entry:

```
1  : wife a lemon : LexicalEntry ;
2    lexinfo : partOfSpeech lexinfo : noun ;
3    lemon : canonicalForm [ lemon : writtenRep "wife"@en ] ;
4    lemon : synBehavior [ rdf : type lexinfo : NounPPFrame ;
5                          lexinfo : copulativeArg       : x_appos ;
6                          lexinfo : prepositionalObject : y_pobj];
7    lemon : sense [ lemon : reference
8                    <http :// dbpedia . org / ontology / spouse >;
9                    lemon : subjOfProp : x_appos ;
10                   lemon : objOfProp  : y_pobj ] .
11
12 : y_pobj lemon : marker [ lemon : canonicalForm
13                          [ lemon : writtenRep "of"@en ] ] .
```

This entry comprises a part of speech (noun), a canonical form (the head noun *wife*), a sense referring to the property `spouse` in the ontology, and a syntactic behavior specifying that the noun occurs with two arguments, a copulative argument that corresponds to the subject of the property and a prepositional object that corresponds to the object of the property and is accompanied by a marker *of*.[9] The specific subcategorization frame is determined by the kind of dependency relations that occur in the pattern. Currently, our approach covers nominal frames (e.g. *activity* and *wife of*), transitive verb frames (e.g *loves*), and adjectival frames (e.g. *Spanish*).

2.5   Lexicalization of Classes

The lexicalization process for classes differs from that for properties in that the corpus is not used. Instead, for each class in the ontology, its label is extracted as lexicalization. In order to also find alternative lexicalizations, we consult WordNet to find synonyms. For example, for the class `http://dbpedia.org/ontology/Activity` with label *activity*, we find the additional synonym *action*, thus leading to the following two entries in the *lemon* lexicon[10]:

---

[9] From a standard lexical point of view the syntactic behavior might look weird. Instead of viewing the specified arguments as elements that are locally selected by the noun, they should rather be seen as elements that occur in a prototypical syntactic context of the noun. They are explicitly named as it would otherwise be impossible to specify the mapping between syntactic and semantic arguments.

[10] As linguistic ontology we use ISOcat (`http://isocat.org`); in the examples, however, we will use the LexInfo vocabulary (`http://www.lexinfo.net/ontology/2.0/lexinfo.owl`) for better readability.

```
1  : activity a lemon : LexicalEntry ;
2    lexinfo : partOfSpeech lexinfo : noun ;
3    lemon : canonicalForm [ lemon : writtenRep " activity "@en ] ;
4    lemon : sense [ lemon : reference
5                    <http :// dbpedia . org / ontology / Activity > ] .
6
7  : action a lemon : LexicalEntry ;
8    lexinfo : partOfSpeech lexinfo : noun ;
9    lemon : canonicalForm [ lemon : writtenRep " action "@en ] ;
10   lemon : sense [ lemon : reference
11                   <http :// dbpedia . org / ontology / Activity > ] .
```

These entries specify a part of speech (noun), together with a canonical form (the class label) and a sense referring to the class URI in the ontology.

3   Evaluation

In this section, we describe the methodology used in our evaluation as well as the evaluation measures, followed by a presentation and discussion of the results. Note that we evaluate our methodology in terms of how well it can support the creation of a lexicon. Of course the extracted patterns could also be used to find new instances of a relation within an information extraction paradigm. However, an evaluation of this potential use is out of the scope of the current paper.

3.1   Methodology and Dataset

We evaluate our approach in two modes: fully *automatic* and *semi-automatic*. In the *automatic mode*, we evaluate the results of our corpus-based lexicon induction method by comparing the automatically generated lexicon with a manually constructed lexicon for DBpedia. The manually constructed lexicon was created by two persons not directly involved in the development and evaluation of the approach presented in this paper. In particular, these lexicon engineers did not have access to the results of the algorithm proposed here when creating their lexica. For the evaluation of our approach in the *semi-automatic mode*, the above mentioned lexicon engineers and one of the authors inspected the automatically generated lexica and added all appropriate lexical entries to their manually created lexicon in case it was appropriate and missing in the lexicon. In this evaluation mode we thus compare the automatically generated lexicon with a superset of the manually constructed lexica. By this, we do not penalize our approach for finding lexical entries that are correct but not contained in the manually constructed lexicon, thus representing a fair evaluation of our approach with respect to the targeted setting in which a lexicon engineer validates the automatically constructed lexical entries.

For the purposes of evaluation, we selected a training set for parameter tuning and a test set for evaluation, each consisting of 10 DBpedia classes and 30 DBpedia properties, in a largely pseudo-random fashion in the sense that we randomly

selected properties from different frequency ranges, i.e. ranging from properties with very few instances to triples with many instances. We then filtered those that turned out to either have no instances—leaving in only one empty property per set, `meltingPoint` and `sublimationPoint`, in order to be able to evaluate possible fallback strategies—or to not have an intuitive lexicalization, e.g. `espnId`. On average, the properties selected for training have 36 100 instances (ranging from 15 to 229 579), while the properties in the test set have 59 532 instances on average (ranging from 9 to 444 025). The training and test sets are also used in the ontology lexicalization task of the QALD-3 challenge[11] at CLEF 2013.

We use the training set to determine the threshold θ, and then evaluate the approach on the unseen properties in the test set.

3.2   Evaluation Measures

For each property, we evaluate the automatically generated lexical entries by comparing them to the manually created lexical entries along two dimensions: i) lexical precision, lexical recall and lexical F-measure, and ii) lexical accuracy. In the first dimension, we evaluate how many of the gold standard entries for a property are generated by our approach (recall), and how many of the automatically generated entries are among the gold standard entries (precision), where two entries count as the same lexicalization if their lemma, part of speech and sense coincide. Thus lexical precision P_{lex} and recall R_{lex} for a property p are defined as follows:

$$P_{lex}(p) = \frac{|entries_{auto}(p) \cap entries_{gold}(p)|}{|entries_{auto}(p)|}$$

$$R_{lex}(p) = \frac{|entries_{auto}(p) \cap entries_{gold}(p)|}{|entries_{gold}(p)|}$$

Where $entries_{auto}(p)$ is the set of entries for the property p in the automatically constructed lexicon, while $entries_{gold}(p)$ is the set of entries for the property p in the manually constructed gold lexicon. The F-measure $F_{lex}(p)$ is then defined as the harmonic mean of $P_{lex}(p)$ and $R_{lex}(p)$, as usual.

The second dimension, lexical accuracy, is necessary in order to evaluate whether the specified subcategorization frame and its arguments are correct, and whether these syntactic arguments have been mapped correctly to the semantic arguments (domain and range) of the property in question. The accuracy of an automatically generated lexical entry l_{auto} for a property p w.r.t. the corresponding gold standard entry l_{gold} is therefore defined as:

$$A_p(l_{auto}) = (frameEq(l_{auto}, l_{gold}) + \frac{|args(l_{auto}) \cap args(l_{gold})|}{|args(l_{gold})|} + \frac{\sum_{a \in args(l_{auto})} map(a)}{|args(l_{auto})|})/3$$

where $frameEq(l_1, l_2)$ is 1 if the subcategorization frame of l_1 is the same as the subcategorization frame of l_2, and 0 otherwise, where $args(l)$ returns the syntactic arguments of l's frame, and where

---

[11] `http://www.sc.cit-ec.uni-bielefeld.de/de/qald`

$$map(a) = \begin{cases} 1, & \textit{if a in } l_{auto} \textit{ has been mapped to the same semantic argument} \\ & \textit{of p as in } l_{\mathrm{gold}} \\ 0, & \textit{otherwise} \end{cases}$$

When comparing the argument mapping of the automatically generated entry with that of the gold standard entry, we only consider the class of the argument, simply being *subject* or *object*. This abstracts from the specific type of subject (e.g. copulative subject) and object (e.g. indirect object, prepositional object, etc.) and therefore allows for an evaluation of the argument mappings independently of the correctness of the frame and frame arguments. The lexical accuracy $A_{lex}(p)$ for a property p is then computed as the average mean of the accuracy values of each generated lexicalization. All measures are computed for each property and then averaged for all properties. In the sections below, we will report only the average values.

3.3   Results and Discussion

Figure 2a shows results for the 30 training properties in automatic mode in terms of P_{lex}, R_{lex}, and F_{lex}, depending on the threshold θ. Accuracy is not plotted, as it is not influenced by θ. Neither are results for the classes, as they also do not vary with θ (recall is 0.73, precision is 0.55, and accuracy 0.9). The value θ is the likelihood that a specific pattern occurs given all the sentences expressing the property in question. On the basis of these results, we identify two θ values that are of interest: a low value around 0.2, which leads to high recall, and a high value around 5.0, which results in a drop in recall but an increase in precision. Having in mind a semi-automatic scenario, in which a lexion engineer validates and, if necessary, corrects the automatically generated lexical entries, we put more emphasis on recall, as it is easier and faster to filter out wrong entries than to discover and add missing one.

Figure 2b gives the results in terms of the average precision P_{lex}, recall R_{lex}, and F-measure F_{lex}, as well as average accuracy A_{lex} for the test set in both evaluation modes, for both relevant θ values, and on all 40 URIs. As with the training set, precision increases and recall decreases for higher θ.

In automatic mode, roughly half of the gold standard entries are generated, usually with a fair precision and accuracy, together with an additional amount of lexicalizations, ranging from 2 to 500, that are not in the gold standard. Of these additional lexicalizations, on average 1.4 entries are correct and were added to the gold standard lexicon. This improved precision and recall roughly by 0.2, accuracy even 0.3 and 0.4.

The property `programmingLanguage` is an example of a proeprty that performs quite bad in terms of precision. Here, six out of seven gold standard lexicalizations are found, leading to a recall of 0.85 and accuracy of 0.96, but also more than 500 wrong lexicalizations are created, yielding a precision of 0.01. The main reason is that the entity labels are not yet preprocessed and therefore take

	θ	R_{lex}	P_{lex}	F_{lex}	A_{lex}
Full	0.2	0.43	0.27	0.24	0.37
	5.0	0.42	0.4	0.34	0.37
Semi	0.2	0.64	0.48	0.41	0.76
	5.0	0.59	0.69	0.55	0.68
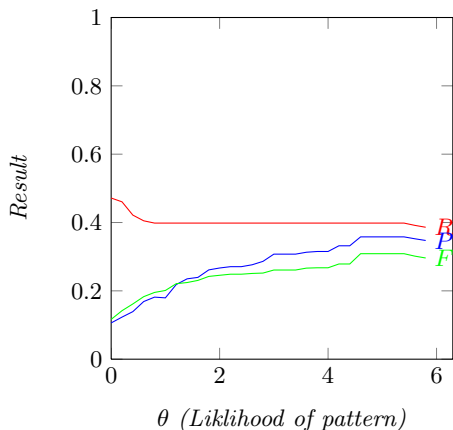
(a) Results on train in automatic mode, with varying θ

(b) Results on test in fully automatic and semi-automatic mode, with fixed θ

Fig. 2. Results on training (a) and test (b) dataset

forms such as *C (programming language)*, which, in combination with the index lookup, leads to the extraction of sentences that might not be relevant, and also hinders the dependency path search between those entities. A similar problem is connected to datatype properties: Literals, such as floating point numbers and dates, have to be preprocessed in order to be found in corpus sentences. Also, the property `elevation`, for example, relates (among others) Barcelona with the number 12, which co-occur in quite some sentences that have nothing to do with the property, such as: *Originally from Barcelona, Spain, he was born March 12, 1971.* A more sophisticated way of filtering patterns found for datatype properties is thus of high importance.

A more general limitation of our tool is that it yet only creates three frame types, so more complex entries such as *to write music for* (lexicalizing the property `musicBy`) still cannot be created. Another problem of the approach is that not all lexicalizations in the gold standard lexicon do occur in the given corpus. For example, for *write music for* no sentence can be found; also no sentences with *sublime* are found that contain an entity pair related by the property `sublimationPoint`.

The average processing time amounts to around 15 seconds per property for the test dataset, assuming that the sentences extracted from the corpus have already been parsed.

4   Related Work

In this section we briefly discuss related work in the area of extracting lexical patterns or paraphrases from corpora that verbalize a given relation in an ontology. An approach that is similar in spirit to our approach is *Wanderlust* [1] which

relies on a dependency parser to find grammatical patterns in a given corpus—
Wikipedia in their case as in ours. These patterns are generic and non-lexical
and can be used to extract any semantic relation. Wanderlust differs from our
approach in several aspects. First, our dependency paths are anchored in par-
ticular lexical entries. Second, we start from a given property and use instance
data to find all different lexical variants of expressing one and the same prop-
erty. Wanderlust, on the other hand, maps each dependency path to a different
property (modulo some postprocessing to detect subrelations) and is in principle
not able to find different variants of expressing one and the same property, so
that semantic normalization is not achieved.

Another related tool is DIRT [7] (*Discovery of Inference Rules from Text*),
which is an unsupervised method for finding inferences in text, so that *x is author
of y* is a paraphrase of *x wrote y*. DIRT relies on a similarity-based approach
to discover similar dependency paths, where two paths are similar if they show
a high degree of overlap in the nouns that appear at the argument positions of
the paths. We could esily extend our approach to also exploit similarity of the
nouns occurring as arguments in patterns to find further paraphrases. The main
difference to our approach is that DIRT does not rely on an existing knowledge
base of instantiated triples to bootstrap the acquisition of patterns from textual
data, thus being completely unsupervised. Given the fact that nowadays there
are large knowledge bases such as Freebase and DBpedia there is no reason why
an approach should not exploit the available instances of a property or class to
bootstrap the acquisition process.

A very similar system, BOA [5], also relies on existing triples from a knowledge
base, in particular DBpedia. BOA applies a recursive procedure, starting with
extracting triples from linked data, then extracting natural language patterns
from sentences and inserting this patterns as RDF data back into the Linked
Data Cloud. However, BOA rely on simple string-based generalization techniques
to find actual patterns. This makes it difficult to discard optional modifiers and
can generate a high amount of noise. This has been corroborated by initial
experiments in our lab on inducing patterns from all the context between the
two entities in question.

Espresso [11] employs a minimally supervised bootstrapping algorithm which,
based on only a few seed instances of a relation, learns patterns to extract more
instances. Espresso is thus comparable to our approach in the sense that both rely
on a set of seed sentences to induces patterns. In our case, these are derived from
a knowledge base, while in the case of Espresso they are manually annotated.
A difference is that we rely on dependency paths connecting two entities, which
yields a principled approach to discarding modifiers and yielding more general
patterns. A system that is similar to Espresso and uses dependencies is the one
proposed by Ittoo & Bouma [6]. In contrast to Espresso, we have not evaluated
our approach on a relation extraction task. A further difference is that Espresso
leverages the web to find further occurrences of the seed instances. The corpus
we use, Wikipedia, is several order of magnitude bigger compared to the corpora
used by Espresso, but nevertheless it would be interesting to extend our approach

to work with web data in order to overcome data sparseness (e.g. as in [3]). This is clearly an option to make use of in case not enough instances are available or not enough seed sentences can be found in the given corpus to bootstrap the pattern acquisition process.

5   Conclusion and Future Work

We presented an approach to the automatic induction of ontology lexica and instantiated it for DBpedia with a corresponding Wikipedia corpus. The approach itself is independent of the chosen domain and could be applied to any other dataset. The results will, however, depend on the specific modelling of the RDF data and the size and quality of the corpus. In particular, our approach faces two principled shortcomings that we want to address in future work. They concern domain and range restrictions as well as verbalizations of complex senses.

First, our approach does not yet check whether patterns are appropriate verbalizations only for a domain or range of the target property. For example, the property `team`, which connects an athlete or manager with a sports team, could be verbalized as *plays for* in case the subject is a football, basketball or volleyball player, as *races for* in case the subject is a cyclist or race driver, and as *manages* if the subject is a sports manager. This can be captured by additionally checking the set of entity pairs that led to a certain pattern for a common subclass of the domain or range of the target property.

Second, our approach only finds verbalizations for simple classes and properties, but not for more complex constructs such as property chains. For example, *born in* is found as verbalization for `dbpedia:birthPlace`, connecting people to the city and sometimes also the country of their birth. This, however, misses the fact that in the dataset the country of birth is not always expressed directly by `dbpedia:birthPlace`, but often indirectly by the property chain `dbpedia:birthPlace ∘ dbpedia:country`. A generated lexicon should contain both senses for *born in*.

Additionally, future work will include an ongoing effort in increasing the number and quality of patterns found. One direction to explore is the extent to which our approach can benefit from the preprocessing of corpus sentences, e.g. by applying reference resolution. Consider, for example, the following sentences: *Barack Obama hosted a White House dinner. He and his personal secretary decided to mainly serve vegan food.* The pattern *and his personal secretary* in the second sentence can only be extracted if the reference of the pronoun *he* is resolved to the entity Barack Obama.

Overall, in this paper we have proposed a first step towards lowering the cost for creating lexica for a given ontology. Such lexica are crucial for any approach requiring access to information about how properties and classes are verbalized in a given language. While our evaluation has shown that our approach is promising, future work will provide a more extensive proof-of-concept, showing that the lexica can be exploited successfully for tasks such as question answering and natural language generation, also in a multilingual settings.

References

1. Akbik, A., Broß, J.: Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In: Proceedings of the Workshop on Semantic Search in Conjunction with the 18th Int. World Wide Web Conference (2009)
2. Bernstein, A., Kaufmann, E., Kaiser, C., Kiefer, C.: Ginseng: A guided input natural language search engine. In: Proceedings of the 15th Workshop on Information Technologies and Systems, pp. 45–50 (2005)
3. Blohm, S., Cimiano, P.: Using the web to reduce data sparseness in pattern-based information extraction. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 18–29. Springer, Heidelberg (2007)
4. Bouayad-Agha, N., Casamayor, G., Wanner, L.: Natural language generation and semantic web technologies. Semantic Web Journal (in press)
5. Gerber, D., Ngomo, A.: Bootstrapping the linked data web. In: Proceedings of the 10th International Semantic Web Conference, ISWC (2011)
6. Ittoo, A., Bouma, G.: On learning subtypes of the part-whole relation: Do not mix your seeds. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1328–1336 (2010)
7. Ling, D., Pantel, P.: DIRT - discovery of inference rules of text. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 323–328. ACM (2001)
8. Lopez, V., Fernandez, M., Motta, E., Stieler, N.: Poweraqua: Supporting users in querying and exploring the semantic web. Semantic Web Journal, 249–265 (2012)
9. McCrae, J., Spohr, D., Cimiano, P.: Linking lexical resources and ontologies on the semantic web with lemon. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 245–259. Springer, Heidelberg (2011)
10. Mellish, C., Sun, X.: The semantic web as a linguistic resource: opportunities for natural language generation. In: Proceedings of 26th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, pp. 298–303. Elsevier (2006)
11. Pantel, P., Pennacchiotti, M.: Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In: Proceedings of the 21st International Conference on Computational Linguistics (COLING), pp. 113–120. ACM (2006)
12. Third, A., Williams, S., Power, R.: OWL to english: a tool for generating organised easily-navigated hypertexts from ontologies. In: Proceedings of 10th International Semantic Web Conference (ISWC), pp. 298–303 (2011)
13. Unger, C., Bühmann, L., Lehmann, J., Ngonga-Ngomo, A.-C., Gerber, D., Cimiano, P.: Sparql template-based question answering. In: Proceedings of the World Wide Web Conference (WWW), pp. 639–648. ACM (2012)
14. Walter, S., Unger, C., Cimiano, P., Bär, D.: Evaluation of a layered approach to question answering over linked data. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part II. LNCS, vol. 7650, pp. 362–374. Springer, Heidelberg (2012)

SQUALL: A Controlled Natural Language as Expressive as SPARQL 1.1

Sébastien Ferré

IRISA, Université de Rennes 1
Campus de Beaulieu, 35042 Rennes cedex, France
`ferre@irisa.fr`

Abstract. The Semantic Web is now made of billions of triples, which are available as Linked Open Data (LOD) or as RDF stores. The most common approach to access RDF datasets is through SPARQL, an expressive query language. However, SPARQL is difficult to learn for most users because it exhibits low-level notions of relational algebra such as union, filters, or grouping. We present SQUALL, a high-level language for querying and updating an RDF dataset. It has a strong compliance with RDF, covers all features of SPARQL 1.1, and has a controlled natural language syntax that completely abstracts from low-level notions. SQUALL is available as two web services: one for translating a SQUALL sentence to a SPARQL query or update, and another for directly querying a SPARQL endpoint such as DBpedia.

1 Introduction

An open challenge of the Semantic Web [12] is *semantic search*, i.e., the ability for users to browse and search semantic data according to their needs. Semantic search systems can be classified according to their *usability*, the *expressive power* they offer, their *compliance* to Semantic Web standards, and their *scalability*. The most expressive approach by far is to use SPARQL [17], the standard RDF query language. SPARQL 1.1[1] features graph patterns, filters, unions, differences, optionals, aggregations, expressions, subqueries, ordering, etc. However, SPARQL is also the least usable approach, as it is defined at a low-level in terms of relational algebra. There are mostly two approaches to make more usable semantic search systems: navigation and natural language. Navigation is used in *semantic browsers* (e.g., Fluidops Information Workbench[2]), and in *semantic faceted search* (e.g., SlashFacet [11], BrowseRDF [16], Sewelis [6]). Semantic faceted search can reach a significant expressiveness, but still much below than SPARQL 1.1, and it does not scale easily to large datasets such as DBpedia[3]. Natural language is used in search engines in various forms, going from full natural language (e.g., FREyA [3], Aqualog [14]) to mere keywords (e.g., NLP-Reduce [13]) through controlled natural languages (e.g., Ginseng [1]). Questions

---

[1] `http://www.w3.org/TR/sparql11-query/`
[2] `http://iwb.fluidops.com/`
[3] `http://dbpedia.org`

in natural language are translated to SPARQL queries, but in general, only a small fragment of SPARQL is used. This means that even if full natural language is allowed, expressiveness is in fact strongly limited. In practice, SPARQL remains the main way to search RDF datasets. A first reason may be that users really need expressiveness in practice, at least when they get specifically interested in a dataset. A second reason may be that most semantic search systems require a lot of preparation before being applied to a specific dataset (e.g., definition of facets or lexicon, derivation of a grammar from an ontology [1]), while SPARQL requires no preparation at all.

A less studied aspect is the update of RDF datasets, i.e., the insertion and deletion of triples. SPARQL 1.1 offers an update language to this purpose. Proposals for more usable interfaces have been made in faceted search (e.g., UTILIS [10]), and in CNL (e.g., ACE [8]). We think that update (and creation) of RDF data is as important as querying because if no data is created, there is nothing to be searched.

In this paper, we present SQUALL, a Semantic Query and Update High-Level Language[4]. Its contribution is to offer an expressiveness that is equivalent to SPARQL 1.1 Query/Update (SPARQL for short), while providing a high-level syntax that completely abstracts from low-level notions such as bindings or relational algebra. SQUALL can be translated to SPARQL, and is therefore fully compliant with Semantic Web standards. In fact, SQUALL qualifies as a Controlled Natural Language (CNL) [19,7]. The main advantage of CNLs is to reuse the cognitive capabilities of people for communicating knowledge, and therefore to reduce the learning effort for using the language. To the best of our knowledge, no existing CNL is strongly compliant with RDF and SPARQL. ACE [7] has its own underlying formalism (Discourse Representation Constructs), and SOS and Rabbit cover OWL ontologies and assume linguistic knowledge [18]. SQUALL does not require any domain-specific linguistic knowledge: e.g., knowing that "person" is a noun, whose plural is "people", and that "knows" is a transitive verb, whose passive is "known". This means that SQUALL is less natural at the lexical level, but that it is applicable to SPARQL endpoints without any preparation.

Section 2 is a short introduction to the Semantic Web and SPARQL. Section 3 describes the different steps that enables the translation from SQUALL sentences to SPARQL queries and updates. Section 4 evaluates the expressiveness of SQUALL by giving for each SPARQL feature its counterpart in SQUALL along with examples. Section 5 evaluates the naturalness of SQUALL on the QALD benchmark. Section 6 concludes the paper.

2   Semantic Web: RDF and SPARQL

The Semantic Web (SW) is founded on several representation languages, such as RDF, RDFS, and OWL, which provide increasing inference capabilities [12].

---

[4] Web forms, examples, and source code can be found from the SQUALL homepage:
`http://www.irisa.fr/LIS/softwares/squall`

The two basic units of these languages are *resources* and *triples*. A resource can be either a URI (Uniform Resource Identifier), a literal (e.g., a string, a number, a date), or a *blank node*, i.e., an anonymous resource. A URI is the absolute name of a *resource*, i.e., an entity, and plays the same role as a URL w.r.t. web pages. Like URLs, a URI can be a long and cumbersome string (e.g., `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`), so that it is often denoted by a qualified name (e.g., `rdf:type`), where `rdf:` is the RDF namespace. In the N3 notation, the default namespace `:` can be omitted for qualified names that do not collide with reserved keywords (*bare qualified names*).

A triple (*s p o*) is made of 3 resources, and can be read as a simple sentence, where *s* is the subject, *p* is the verb (called the predicate), and *o* is the object. For instance, the triple (`Bob knows Alice`) says that "Bob knows Alice", where `Bob` and `Alice` are the bare qualified names of two individuals, and `knows` is the bare qualified name of a property, i.e., a binary relation. The triple (`Bob rdf:type man`) says that "Bob has type man", or simply "Bob is a man". Here, the resource `man` is used as a class, and `rdf:type` is a property from the RDF namespace. The triple (`man rdfs:subClassOf person`) says that "man is a subclass of person", or simply "every man is a person". The set of all triples of a knowledge base forms an RDF graph.

Query languages provide on semantic web knowledge bases the same service as SQL on relational databases. They generally assume that implicit triples have been inferred and added to the base. The standard RDF query language, SPARQL, reuses the `SELECT FROM WHERE` shape of SQL queries, using graph patterns in the `WHERE` clause. A graph pattern G is one of:

- a triple pattern (*s p o* .) made of RDF terms and variables (e.g., `?x`),
- a join of two patterns ($\{G_1 \ G_2\}$),
- an union of two patterns (G_1 `UNION` G_2),
- an optional pattern (`OPTIONAL` G_1),
- a filter pattern (`FILTER` C), where C is a constraint, i.e., either a Boolean expression based on primitive predicates (e.g., comparison, string matching), or a negated graph pattern (`NOT EXISTS` G_1),
- a named graph pattern (`GRAPH` g G_1), where g is URI or a variable denoting a named graph,
- a subquery.

Aggregations and expressions can be used in the `SELECT` clause (e.g., `COUNT`, `SUM`, `2 * ?x`), and `GROUP BY` clauses can be added to a query. Solution modifiers can also be added to the query for ordering results (`ORDER BY`) or returning a subset of results (`OFFSET`, `LIMIT`). Other query forms allow for closed questions (`ASK`), for returning the description of a resource (`DESCRIBE`), or for returning RDF graphs as results instead of tables (`CONSTRUCT`). SPARQL has been extended into an update language to insert and delete triples in/from a graph. The most general update form is `DELETE` D `INSERT` I `WHERE` G, where I and D must be sets of triple patterns, and G is a graph pattern that defines bindings for variables occuring in I and D.

3   Translation from SQUALL to SPARQL

The idea behind SQUALL is to offer a high-level substitute of SPARQL. This implies that all of SQUALL should be translatable to SPARQL, and that all of SPARQL should be expressible in SQUALL. We do not have yet a formal proof of the latter, but all features of SPARQL are covered in SQUALL, up to a few minor exceptions, as shown in Section 4. The implementation of SQUALL as a translator to SPARQL is an obvious choice as it leverages existing work on efficient SPARQL query engines, and satisfies interoperability with existing RDF stores, which provide access through SPARQL endpoints. We now describe the different steps of such a translation.

3.1   Lexical Analysis

In the current implementation of SQUALL, there are no proper lexical analysis because we use the same lexical conventions as in SPARQL, plus bare qualified names like in N3 (see Section 2). This comes from our choice to make SQUALL directly applicable to RDF datasets without preparation. Of course, if linguistic knowledge is available for the resources of the datasets (e.g., "actor", "stars", "starring" all refer to the property `dbpedia:starring`), then a preprocessing stage may be applied on SQUALL sentences to allow for more natural sentences at the lexical level. Fortunately, namespaces and bare qualified names allow for relatively natural sentences, as shown in examples in this paper and on the Web page. For example, DBpedia uses three namespaces: one for the ontology (classes and properties), another for additional properties, and a last one for individual resources. By associating to them respectively the prefixes :, `dbp:`, `res:`, we can write in SQUALL "Which Film has director res:Tim_Burton ?". `Film` stands for the URI `http://dbpedia.org/ontology/Film`, and `res:Tim_Burton` stands for the URI `http://dbpedia.org/resource/Tim_Burton`. In SQUALL, classes can be used as nouns and intransitive verbs, properties can be used as relation nouns and transitive verbs, and resources can be used as proper nouns.

3.2   Syntactic and Semantic Analysis

The syntactic and semantic analysis of SQUALL are formally defined and implemented as a Montague grammar made of around 100 rules[5]. Montague grammars [4] are an approach to natural language semantics that is based on formal logic and λ-calculus. It is named after the American logician Richard Montague, who pioneered this approach [15]. A Montague grammar is a context-free generative grammar, where each rule is decorated by a λ-term that denotes the semantics of the syntactic construct defined by the rule. The semantics is defined in a fully compositional style, i.e., the semantics of a construct is always

---

[5] The full Montague grammar can be found in the source code at `https://bitbucket.org/sebferre/squall2sparql/src` (file `syntax.ml`), or in a previous paper [5] for an earlier version of SQUALL.

a composition of the semantics of sub-constructs. The obtained semantics for a valid SQUALL sentence serves as an intermediate representation before generation to possibly different target languages, here SPARQL.

SQUALL sentences are decomposed into noun phrases, verb phrases, relatives, determiners, prepositional phrases. They can express assertions when ending with a full stop (e.g., "res:Paris is the capital of res:France.") or questions when ending with a question mark (e.g., "What is the capital of res:France?"). So far, anaphoras are handled with variables (e.g., ?X), but those are rarely needed. As an illustration, we consider a complex sentence that covers many features of SQUALL: "For which researcher-s ?X, in graph DBLP every publication whose author is ?X and whose year is greater than 2000 has at least 2 author-s?". Its syntactic analysis is (see [5] for details)

"$[_S$for $[_{NP}[_{Det}$which$]$ $[_{NG1}[_{P1}$researcher-s$]$ $[_{AR}[_{App}$?X$]]]],$ $[_S[_{PP}$in $[_{Prep}$graph$]$ $[_{NP}$DBLP$]]$ $[_S[_{NP}[_{Det}$every$]$ $[_{NG1}[_{P1}$publication$]$ $[_{AR}[_{Rel}[_{Rel}$whose $[_{NG2}[_{P2}$author$]]$ $[_{VP}$is $[_{NP}$?X$]]]$ and $[_{Rel}$whose $[_{NG2}[_{P2}$year$]]$ $[_{VP}$is $[_{Rel}$greater than $[_{NP}$2000$]]]]]]]$ $[_{VP}$has $[_{Det}$at least 2$]$ $[_{P2}$author-s$]]]]]]]$".

3.3   SPARQL Generation

The last step is the generation of a SPARQL query or update from the intermediate representation of semantics. It is much simpler than syntactic and semantic analysis (around 100 lines of code) because it mostly consists in mapping logical constructs to SPARQL constructs, which are at the same level of abstraction. Note that the intermediate representation makes it easy to support another target query language, e.g., Datalog [2]. As an illustration, the SPARQL translation of the above example is as follows:

```
SELECT DISTINCT ?X WHERE {
    ?X a :researcher .
    FILTER NOT EXISTS {
        GRAPH :DBLP {
            ?x3 a :publication .
            ?x3 :author ?X .
            ?x3 :year ?x6 .
            FILTER (?x6 > 2000) . }
        FILTER NOT EXISTS {
            GRAPH :DBLP {
                { SELECT DISTINCT ?x3 (COUNT(?x9) AS ?x7)
                  WHERE { ?x3 :author ?x9 . }
                  GROUP BY ?x3 }
                FILTER (2 <= ?x7) . } } } }
```

The two nested `FILTER NOT EXISTS` encode the universal quantifier "every", and the subquery with aggregation encodes the numeric quantifier "at least 2".

3.4   Implementation as a Web Service

SQUALL is available as two Web services. A *translation form* takes a SQUALL sentence and returns its SPARQL translation. A *query form* takes a SPARQL

endpoint URL, namespace definitions, and a SQUALL sentence, sends the SPARQL translation to the endpoint, which returns the list of answers to the query. SQUALL is also available as a command line tool that can be called from scripts or programs locally. SQUALL is implemented in about 2000 lines of OCaml[6], a functional language where Montague grammars have a natural encoding. The source code is available as a BitBucket repository from the SQUALL homepage.

4   Expressiveness Compared to SPARQL

We evaluate the expressiveness of SQUALL by giving for each SPARQL feature its counterpart in SQUALL. This list of features is adapted and extended from a comparison of RDF query languages [9]. For each feature, SQUALL sentences are given as illustrations. For the sake of simplicity, we assume that all resources belong to a same namespace so that bare qualified names can be used (e.g., "person", "author", "NLDB"). The SPARQL translation of SQUALL sentences can be obtained from the translation form at `http://lisfs2008.irisa.fr/ocsigen/squall/`.

Triple Patterns. Each noun or non-auxiliary verb plays the role of a class or a predicate in a triple pattern. If a question is about a class or a predicate, the verbs "belongs" and "relates" are respectively used.

- "Which person is the author of a publication whose publication_year is 2012?"
- "To which nationality does John_Smith belong?" (here, "nationality" is a meta-class whose instances are classes of persons: e.g., "French", "German").
- "What relates John_Smith to Mary_Well?"

Updates. Updates are obtained by sentences in the affirmative. A sequence of affirmative sentences generates a sequence of updates.

- "Paper42 has author John_Smith and has publication_year 2012."
- "John_Smith know-s Mary_Well. Mary_Well know-s John_Smith."

Queries. SELECT queries are obtained by open questions, using one or several question words ("which" as a determiner, "what" or "who" as a noun phrase). Queries with a single selected variable can also be expressed as imperative sentences. ASK queries are obtained by closed questions, using either the word "whether" in front of an affirmative sentence, or using auxiliary verbs and subject-auxiliary inversion.

- "Which person is the author of which publication?"
- "Give me the author-s of Paper42."
- "Whether John_Smith know-s Mary_Well?"
- "Does Mary_Well know the author of Paper42?"

---

[6] `http://caml.inria.fr/ocaml/`

Solution Modifiers. The ordering of results (`ORDER BY`) and partial results (`LIMIT`, `OFFSET`) are expressed with adjectives like "highest", "2nd lowest", "10 greatest".

- "Which person-s have the 10 greatest age-s?"
- "What are the author-s of the publication-s whose publication_year is the 2nd latest?"

Built-ins. Built-in functions and operators used in SPARQL filters and expressions are expressed by pre-defined nouns, verbs, and relational adjectives: e.g., "month", "contains", "greater than". They can therefore be used like classes and properties.

- "Which person has a birth_date whose month is 3 and whose year is greater than 2000?"
- "Give me the publication-s whose title contains "natural language"?"

Join. The coordination "and" can be used with all kinds of phrases. It generates complex joins at the relational algebra level.

- "John_Smith and Mary_Well have age 42 and are an author of Paper42 and Paper43."

Union. Unions of graph patterns are expressed by the coordination "or", which can be used with all kinds of phrases, like "and".

- "Which teacher or student teach-es or attend-s a course whose topic is NL or DB?"

Option. Optional graph patterns are expressed by the adverb "maybe", which can be used in front of all kinds of phrases, generally verb phrases.

- "The author-s of Paper42 have which name and maybe have which email?"

Negation. The negative constraint on graph patterns (`NOT EXISTS`) is expressed by the adverb "not", which can be used in front of all kinds of phrases, and in combination with auxiliary verbs. In updates, negation entails the deletion of triples.

- "Which author of Paper42 has not affiliation Salford_University?"
- "John_Smith is not a teacher and does not teach Course101."

Quantification. Quantifiers have no direct counterpart in SPARQL, and can only be expressed indirectly with negation or aggregation. In SQUALL, they are expressed by determiners like "a", "every", "no", "some", "at least 3", "the most", "the". The latter "the" is interpreted existentially in queries, and universally in updates. The universal quantifier in updates allow for batches of updates, and correspond to the use of a `WHERE` clause in SPARQL updates.

- "Every author of Paper42 has affiliation the university whose location is Salford."
- "Every author of which publication has affiliation Salford_University?"
- "Which person-s are the author of the most publication-s?"

Aggregation and Grouping. Aggregation is expressed by the question determiner "how many", by relational nouns such as "number", "sum", "average", and by adjectives such as "total", "average". Grouping clauses are introduced by the word "per".

- "How many publication-s have author John_Smith?"
- "What is the number of publication-s per author?"
- "What is the average age of the author-s of Paper42?"

Expressions. Operators and functions are defined as coordinations so that they can be applied on different kinds of phrases: e.g., relational nouns, noun phrases.

- "Which publication has the lastPage - the firstPage greater than 10?"
- "Return concat(the firstname, " ", the lastname) of all author-s of Paper42."

Property Paths. Property sequences and inverse properties are covered by the flexible syntax of SQUALL. Alternative and negative paths are respectively covered by the coordination "or" and the adverb "not". Reflexive and transitive closures of properties have no obvious linguistic counterpart, and are expressed by property suffixes among "?", "+", and "*".

- "Which publication-s cite+ Paper42?" (i.e., *Which publications cite Paper42 or cite a publication that cite Paper42, etc?*)

Named Graphs. The GRAPH construct of SPARQL, which serves to restrict graph pattern solutions to a named graph, can be expressed using "in graph" as a preposition. A prepositional phrase can be inserted at any location in a sentence, and its scope is the whole sentence.

- "Who is the author of the most publication-s in graph Salford_Publications?"
- "In which graph is John_Smith the author of at least 10 publication-s?"

Graph Literals. The SPARQL query forms CONSTRUCT and DESCRIBE return graphs, i.e. sets of triples, instead of sets of solutions. A DESCRIBE query is expressed by the imperative verb "describe" followed by a resource or a universally-quantified noun phrase. A CONSTRUCT query is expressed by using curly brackets to quote sentences and make them a graph literal.

- "Describe the author-s of Paper42."
- "For every person ?X that is an author of a publication that has author a person ?Y that is not ?X, return { ?X has coauthor ?Y and ?Y has coauthor ?X. }."

A detailed review of SPARQL 1.1 grammar reveals only a few missing features: (1) updates at graph level (e.g., LOAD, DROP), (2) use of results from other endpoints (e.g., VALUES, SERVICE), (3) transitive closure on complex property paths (e.g., (^author/author)+ for co-authors of co-authors, and so on).

Table 1. Comparison of the average length of questions in the three languages

language	natural language	SQUALL	SPARQL
average question length	45	55	173

5   Naturalness Evaluation on the QALD Challenge

The QALD[7] challenge (Query Answering over Linked Data) provides "a benchmark for comparing different approaches and systems that mediate between a user, expressing his or her information need in natural language, and semantic data". The last campaign, QALD-2, provides hundreds of questions in natural language over two datasets: DBpedia and MusicBrainz. The principle of the challenge is that a training set of 100 questions is provided, along with SPARQL translations and answers, and systems are evaluated on a test set that is made of 100 new questions. Systems are compared in terms of precision and recall for the test questions. Here, we do not measure precision and recall because SQUALL is not a system that produces answers from natural language questions, but a language that can be used to express queries. Because SQUALL has the same expressiveness as SPARQL (see Section 4), it is possible to reach perfect precision and recall. The question we try to answer in this section is: *How close to natural language questions are the SQUALL sentences, when made equivalent to the SPARQL queries?* To this purpose, we here focus on the 100 questions of the training set for the DBpedia dataset, from the QALD-2 campaign. The 100 questions of the test set are very similar, and therefore they do not add to our evaluation. The SQUALL version of those 200 questions are available from the example page of the SQUALL page. For each question, the SPARQL translation and answers from DBpedia can be obtained in two clicks.

The Concision of SQUALL is Comparable to Natural Language. Table 1 compares the average length of questions in three languages: natural language (original QALD question), SQUALL (our version of questions), SPARQL (the golden standard provided by QALD organizers). Whereas SPARQL queries are nearly four times longer than natural language questions, SQUALL queries are only about 20% longer. The difference between natural language and SQUALL is largely explained by the namespaces in qualified names (e.g., `res:IBM` instead of `IBM`).

SQUALL Queries Look Natural. The use of variables is hardly ever necessary in SQUALL (none was used in the 100 training questions), while SPARQL queries are cluttered with many variables. No special notations were used, except for namespaces. Only grammatical words are used to provide syntax, and they are used like in natural language. There are 9 out of 100 questions where SQUALL is identical to natural language, up to proper names which are replaced by URIs:

- "Is res:Proinsulin a Protein?"
- "What is the currency of res:Czech_Republic?"

---

[7] `http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/`

- "What is the areaCode of res:Berlin ?"
- "Who is the owner of res:Universal_Studios?"
- "What are the officialLanguage-s of res:Philippines?"
- "What is the highestPlace of res:Karakoram?"
- "Give me the foaf:homepage of res:Forbes?"
- "Give me all yago:SchoolTypes."
- "Which Country has the most dbp:officialLanguages?"

Most Discrepancies between Natural Language and SQUALL are a Matter of Vocabulary. Most discrepancies come from the fact that for each concept, a single word has been chosen in the DBpedia ontology, and related words are not available as URIs. Because SQUALL sentences use URIs as nouns and verbs, some reformulation is necessary. In the simplest case, it is enough to replace a word by another: e.g., "wife" vs "dbp:spouse". In other cases, a verb has to be replaced by a noun, which requires changes in the syntactic structure: e.g., "Who developed the video game World of Warcraft?" vs "Who is the developer of res:World_of_Warcraft?". An interesting example is "Who is the daughter of Bill Clinton married to?" vs "Who is the dbp:spouse of the child of res:Bill_Clinton?". The former question could be expressed in SQUALL if "marriedTo" was made an equivalent property to "dbp:spouse", and if "daughter" was made a subproperty of "child". In fact, this kind of discrepancy could be resolved, either by enriching the ontology with related words, or by preprocessing SQUALL sentences using natural words to replace them by URIs. The latter solution has already been studied as a component of existing question answering systems [3,14], and could be combined with translation from SQUALL to SPARQL.

Some Discrepancies are Deeper in that they Exhibit Conceptual Differences between Natural Language and the Ontology. We shortly discuss three cases:

- "List all episodes of the first season of the HBO television series The Sopranos!" vs "List all TelevisionEpisode-s whose series is res:The_Sopranos and whose seasonNumber is 1.". In natural language, an episode is linked to a season, which in turn is linked to a series. In DBpedia, an episode is linked to a series, on one hand, and to a season number, on the other hand. In DBpedia, a season is not an entity, but only an attribute of episodes.
- "Which caves have more than 3 entrances?" vs "Which Cave-s have an dbp:entranceCount greater than 3?". The natural question is nearly a valid sentence in SQUALL, but it assumes that each cave is linked to each of its entrances. However, DBpedia only has a property "dbp:entranceCount" from a cave to its number of entrances.
- "Which classis does the Millepede belong to?" vs "What is the dbp:classis of res:Millipede?". The natural question is again a valid SQUALL sentence (after moving 'to' at the beginning), but it assumes that res:Millipede is an instance of a class, which is itself an instance of dbp:classis. DBpedia does not define classes of classes, and therefore uses dbp:classis as a property from a species to its classis.

Those discrepancies are more difficult to solve. A first solution would be to make the ontology better fit usage in natural language. A second solution is to reformulate a natural question so that it matches the ontology.

6   Discussion and Conclusion

In the spectrum that goes from full natural language to formal languages like SPARQL or SQL, SQUALL (Semantic Query and Update High-Level Language) occupies a unique position. It offers the same expressiveness as SPARQL for querying and updating RDF data, and still qualifies as a controlled natural language (CNL). This means that among the natural language interfaces, SQUALL is the one that is by far the most expressive; and that among the formal languages, SQUALL is the one that is the most natural. The limit of SQUALL is that end-users have to comply with its controlled syntax, and have to know the RDF vocabulary (i.e., *Which are the classes and properties?*). However, the important result is that SQUALL can be used as a substitute for SPARQL because this entails no loss, neither in expressiveness, nor in precision.

SQUALL can be used as a front-end language when no linguistic knowledge is available about an RDF dataset, exactly like for SPARQL. As future work, SQUALL could also be used as an intermediate language, combining it with existing work in natural language interfaces. As discussed in Section 5, most discrepancies between SQUALL and spontaneous natural language are related to vocabulary and ontology. Interestingly, most of existing work have precisely focused on mapping from words to URIs and reformulation (e.g., Lemon[8]). The other way round, SQUALL provides a rich and flexible grammar (e.g., coordinations on all kinds of phrases, quantification, aggregation), and completely abstracts over low-level aspects of SPARQL (e.g., relational algebra). We therefore think that SQUALL and those existing work, while already useful individually, could strongly benefit from each other.

Future works will address (1) the full coverage of SPARQL 1.1, and its proof by implementing a translation from SPARQL to SQUALL; (2) the guided construction of SQUALL sentences with query-based faceted search [6]; and (3) the use of lexicons for more natural sentences.

References

1. Bernstein, A., Kaufmann, E., Kaiser, C.: Querying the semantic web with Ginseng: A guided input natural language search engine. In: Work. Information Technology and Systems, WITS (2005)
2. Ceri, S., Gottlob, G., Tanca, L.: What you always wanted to know about datalog (and never dared to ask). IEEE Trans. Knowl. Data Eng. 1(1), 146–166 (1989)
3. Damljanovic, D., Agatonovic, M., Cunningham, H.: Identification of the question focus: Combining syntactic analysis and ontology-based lookup through the user interaction. In: Language Resources and Evaluation Conference (LREC). ELRA (2010)

---

[8] http://lemon-model.net/index.html

4. Dowty, D.R., Wall, R.E., Peters, S.: Introduction to Montague Semantics. D. Reidel Publishing Company (1981)
5. Ferré, S.: SQUALL: a controlled natural language for querying and updating RDF graphs. In: Kuhn, T., Fuchs, N.E. (eds.) CNL 2012. LNCS, vol. 7427, pp. 11–25. Springer, Heidelberg (2012)
6. Ferré, S., Hermann, A.: Reconciling faceted search and query languages for the Semantic Web. Int. J. Metadata, Semantics and Ontologies 7(1), 37–54 (2012)
7. Fuchs, N.E., Kaljurand, K., Schneider, G.: Attempto Controlled English meets the challenges of knowledge representation, reasoning, interoperability and user interfaces. In: Sutcliffe, G., Goebel, R. (eds.) FLAIRS Conference, pp. 664–669. AAAI Press (2006)
8. Fuchs, N.E., Schwitter, R.: Web-annotations for humans and machines. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 458–472. Springer, Heidelberg (2007)
9. Haase, P., Broekstra, J., Eberhart, A., Volz, R.: A comparison of RDF query languages. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 502–517. Springer, Heidelberg (2004)
10. Hermann, A., Ferré, S., Ducassé, M.: An interactive guidance process supporting consistent updates of RDFS graphs. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS (LNAI), vol. 7603, pp. 185–199. Springer, Heidelberg (2012)
11. Hildebrand, M., van Ossenbruggen, J., Hardman, L.: /facet: A browser for heterogeneous semantic web repositories. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 272–285. Springer, Heidelberg (2006)
12. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
13. Kaufmann, E., Bernstein, A.: Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. J. Web Semantics 8(4), 377–393 (2010)
14. Lopez, V., Uren, V., Motta, E., Pasin, M.: Aqualog: An ontology-driven question answering system for organizational semantic intranets. Journal of Web Semantics 5(2), 72–105 (2007)
15. Montague, R.: Universal grammar. Theoria 36, 373–398 (1970)
16. Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for RDF data. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 559–572. Springer, Heidelberg (2006)
17. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 30–43. Springer, Heidelberg (2006)
18. Schwitter, R., Kaljurand, K., Cregan, A., Dolbear, C., Hart, G.: A comparison of three controlled natural languages for OWL 1.1. In: Clark, K., Patel-Schneider, P.F. (eds.) Workshop on OWL: Experiences and Directions (OWLED), vol. 258. CEUR-WS (2008)
19. Smart, P.: Controlled natural languages and the semantic web. Tech. rep., School of Electronics and Computer Science University of Southampton (2008), http://eprints.ecs.soton.ac.uk/15735/

Evaluating Syntactic Sentence Compression for Text Summarisation

Prasad Perera and Leila Kosseim

Dept. of Computer Science & Software Engineering
Concordia University
Montreal, Canada
`p_perer@encs.concordia.ca,kosseim@encs.concordia.ca`

Abstract. This paper presents our work on the evaluation of syntactic based sentence compression for automatic text summarization. Sentence compression techniques can contribute to text summarization by removing redundant and irrelevant information and allowing more space for more relevant content. However, very little work has focused on evaluating the contribution of this idea for summarization. In this paper, we focus on pruning individual sentences in extractive summaries using phrase structure grammar representations. We have implemented several syntax-based pruning techniques and evaluated them in the context of automatic summarization, using standard evaluation metrics. We have performed our evaluation on the TAC and DUC corpora using the BlogSum and MEAD summarizers. The results show that sentence pruning can achieve compression rates as low as 60%, however when using this extra space to fill in more sentences, ROUGE scores do not improve significantly.

1 Introduction

Text compression has several practical applications in natural language processing such as text simplification [1], headline generation [2] and text summarization [3]. The goal of automatic text summarization is to produce a shorter version of the information contained in a text collection and produce a relevant summary [4]. In extractive summarization, sentences are extracted from the document collection and assigned a score according to a given topic/query relevance [5] or some other metric to determine how important it is to the final summary. Summaries are usually bound by a word or sentence limit and within these limits, the challenge is to extract and include as much relevant information as possible. However, since the sentences are not processed or modified, they may contain phrases that are irrelevant or may not contribute to the targeted summary. As an example, consider the following topic, query and sentence (1) [1],

Topic: *Southern Poverty Law Center*

Query: *Describe the activities of Morris Dees and the Southern Poverty Law Center*

---

[1] All examples are taken from the TAC 2008 or DUC 2007 corpora.

(1) *Since co-founding the Southern Poverty Law Center in 1971, Dees has wielded the civil lawsuit like a buck knife, carving financial assets out of hate group leaders who inspire followers to beat, burn and kill.*

In sentence (1), some phrases could be dropped without losing much information relevant to the query. Possible shorter forms of the sentence include :

(1c1) *Since co-founding the Southern Poverty Law Center in 1971, Dees has wielded the civil lawsuit ~~like a buck knife~~, carving financial assets out of hate group leaders who inspire followers to beat, burn and kill.*

(1c2) *~~Since co-founding the Southern Poverty Law Center in 1971,~~ Dees has wielded the civil lawsuit like a buck knife, carving financial assets out of hate group leaders who inspire followers to beat, burn and kill.*

(1c3) *~~Since co-founding the Southern Poverty Law Center in 1971,~~ Dees has wielded the civil lawsuit ~~like a buck knife,~~ carving financial assets out of hate group leaders ~~who inspire followers to beat, burn and kill.~~*

In principle, sentence compression should improve automatic extractive text summarization by removing redundant and less relevant information within sentences and thus preserve space to include more useful information into length-limited summaries. However to the best of our knowledge, very little previous work has focused on measuring the contribution of specific sentence compression techniques as a means to improve summary content.

2   Previous Work on Sentence Compression

Previous sentence compression methods have relied on different techniques ranging from machine learning and classifier based (eg. [3]), syntactic pruning (based on complete parses or shallow parses) (eg. [6,7,8,9]) to keyword based (eg. [10]) techniques.

One early approach to sentence pruning focused on removing inessential phrases in extractive summaries based on an analysis of human written abstracts [9,6]. In their work, the authors have used a syntactic parser to identify different types of phrases which are present in the original sentences but not in human written simplified sentences. These phrases were used to train a Naive Bayes Classifier to decide how likely a phrase is to be removed from a sentence. For evaluation, they have compared their compressed sentences to those compressed by humans and acheived a 78.1% overall success rate but have noted a low success rate for removing adjectives, adverbs and verb phrases. However, the effect on summary content was not indicated.

Another interesting work is that of [3] who proposed a noisy channel model technique based on the hypothesis that there exists a shorter original sentence (s) and the existing longer sentence (t) was formed by adding optional phrases. Given the long string t and every pair of (t, s), the probability $P(t \mid s)$ represents

the likelihood of arriving at the long string t, when s is expanded. Their model was designed considering two key features: preserving grammaticality and preserving useful information. In order to calculate the probabilities, they have used context free grammar parses of sentences and a word based bi-gram estimation model. They have evaluated their system using the Ziff-Davis corpus and have showed that their approach could score similar compression rates compared to human written compressed texts but importance and grammaticality are slightly lower than human-written texts. On the other hand, [11] introduced semantic features to improve a decision tree based classification. Here, the authors used Charniak's parser [12] to generate syntactic trees and incorporated semantic information using WordNet [13]. The evaluation showed a slight improvement in importance of information preserved in shortened sentences. But again, the effect on summarization was not noted. [14] points out that text compression could be seen as a problem of finding a global optimum by considering the compression of the whole text/document. The authors used syntactic trees of each pairs of long and short sentences to define rules to deduce shorter syntactic trees out of original syntactic trees. They also used the Ziff-Davis corpus for their evaluation as well as human judgment. They evaluated their technique based on importance and grammaticality of sentences and the results were lower compared to the scores of the human written abstractions. Similarly, [15] describes the use of integer linear programming model to infer globally optimal compressions while adhering to linguistically motivated constraints and show improvement in automatic and human judgment evaluations. [16] have also described an approach on syntactic pruning based on trasformed dependency trees and a linear integer model. The authors have transformed the dependency trees into graphs where the nodes represents nouns and verbs and these transformed dependency trees are trimmed based on the results of an integer linear programming model that decides the importance of each subtree. Their evaluation has shown an improvement compared to the language model based compression techniques.

The previous work described above were evaluated intrinsically by comparing their results to human generated summmaries. A few previous work did however measure sentence compression extrinsically for the purpose of text summarization. In particular, [10] took a conservative approach and used a list of keyword phrases to identify less significant parts of the text and remove them from long sentences. The keyword list was implemented in an adhoc fashion and was used to omit specific terms. They have evaluated their pruning techniques within their summarization system CLASSY [17] with DUC 2005 [18], and showed an improvement in ROUGE scores. In their participation to the DUC 2006 [19] automatic summarization track, their system placed among the top three based on ROUGE scores.

In contrast, [7] used complete dependency parses and applied pruning rules based on grammatical structures. They used specific grammatical filters including prepositional complements of verbs, subordinate clauses, noun appositions and interpolated clauses. They have achieved a compression rate of 74% while retaining grammaticality and readability of text. In [20] the authors also used

syntactic structures and applied linguistically motivated filtering to simplify sentences. Using the TIPSTER [21] corpus, they identified syntactic patterns which were absent from human-written summaries compared to the original corpus and defined a trimming algorithm consisting of removing sub-trees of grammatical phrase structures while traversing through a complete parsed tree structure. They have evaluated their pruning technique on the DUC 2003 summarization task and showed an improvement in ROUGE scores compared to uncompressed length-limited summaries. Finally, [8] describes the sentence compression module of their text summarization system, based on syntactic level sentence pruning. They have implemented a module of compression which filters adverbial modifiers and relative clauses from original sentences to achieve text compression. Their evaluations were performed using the DUC 2007 summarization track and have showed an improvement in ROUGE scores after applying their compression technique to their summarization system.

As described above, most previous work have evaluated their sentence compression technique intrinsically against human generated compressed sentences. Very few (notably [7,8,20,10]) have evaluated them extrinsically as part of a summarization system but the exact contribution of each technique to the summary content has not been measured.

3 Pruning Heuristics

To evaluate syntactic sentence pruning methods in the context of automatic text summarization, we have implemented several syntax-based heuristics and have evaluated them with standard summarization benchmarks. We took as input a list of extracted sentences ranked by their relevance score as generated by an automatic summarizer. We then performed a complete parse of these sentences, and applied various syntax-based pruning approaches to each tree node to determine whether to prune or not a particular sub-tree. The pruned sentences were then included in the final summary in place of the original sentences and evaluated for content against the given model summaries. Three basic sentence compression approaches were attempted: syntax-driven pruning, syntax and relevancy based pruning, and relevancy-driven syntactic pruning. Let us describe each approach in detail.

3.1 Syntax-Driven Pruning

Our first approach to sentence pruning was based solely on syntactic simplifications. After parsing the extracted sentences deemed relevant by the summarizer, we tried to remove specific sub-trees regardless of their computed relevance to the query/topic. Here, the rationale was that specific syntactic structures by default carry secondary informative content, hence removing them should not decrease the content of the summary significantly. These pruning heuristics are based on the work of [20] and [7] (adapted to English). Specially, we removed: relative clauses, adjective and adverbial phrases, conjuncted clauses as well as specific types of prepositional phrases. Let us describe each heuristic:

Pruning Relative Clauses. A relative clause modifies a noun or noun phrase and is connected to the noun by a relative pronoun, a relative adverb, or a zero relative. As such, they act as adjectival phrases that provide additional information about the noun it modifies. As an example, consider the following sentence:

(2) *"It's over", said Tom Browning, an attorney for Newt Gingrich,*
     who was not present at Thursday's hearing.

Pruning the sub tree structure headed by *who*, which represents a relative clause, results in a shortened sentence.

Pruning Adjective Phrases. An ajective phrase is a word, phrase, or sentence element that enhances, limits or qualifies the meaning of a noun phrase. As complementary phrases, they can often be dropped from a sentence without loosing the main content of the sentence. As an example:

(3) *Mark Barton, the 44-year-old day trader at the center of Thursday's bloody*
     rampage, was described by neighbors in the Atlanta suburb of Morrow as a
     quiet, churchgoing man who worked all day on his computer.

The phrases *44-year-old*, *bloody* and *quiet, churchgoing* are suitable candidates to be pruned from the original phrase structure.

Pruning Adverbial Phrases. An adverbial phrase is a word, phrase, or sentence element that modifies a verb phrase. For example:

(4) *So surely there will be a large number of people who only know us for Yojimbo.*

Here, the phrases *surely* and *only* provide additional information regarding their associated verb phrases, but can often be dropped without affecting the content of the sentence significantly.

Pruning Trailing Conjuncted Verb Phrases. Conjunctions may be used to attach several types of phrases. In our corpora, verb phrases (VPs) are often conjuncted and the second VP is typically shorter and contains secondary information. For example, consider the following sentence:

(5) *The Southern Poverty Law Center has accumulated enough wealth in recent*
     years to embark on a major construction project
     and to have assets totaling around $100 million.

Based on our corpus analysis, we developed a heuristic that removes trailing conjuncted VPs.

Pruning Prepositional Phrases. Prepositional phrases (PP) are used to modify noun phrases, verb phrases or complete clauses. Pruning PPs can be done, but with caution. Indeed, some PPs do contain secondary information which can be removed without hindering the grammar or the semantics of the sentence; while other types of PPs do contain necessary information. Consider the following example:

(6)   *In the Public Records Office in London archivists are creating a catalog*
      of all British public records regarding the Domesday Book of the 11th century.

Here, the prepositional phrase, *In the Public Records Office in London* is attached
to the entire clause; while, *of all British public records* and *of the 11th century* are
attached to the nouns *catalog* and *Domesday Book*. PPs attached to NPs often act
as noun modifiers and as a consequence can be pruned like any adjective phrase.
In addition, PPs attached to an entire clause often present complementary infor-
mation that can also be removed. On the other hand, PPs can be attached to verb
phrases, as in:

(7)   *Australian Prime Minister John Howard today defended the governments*
      decision to go ahead with uranium mining
      on development and environmental grounds.

where *with uranium mining* and *on development and environmental grounds* are
attached to *go ahead*. PPs that modify verb phrases should be pruned with cau-
tion as they may be part of the verb's frame and required to understand the verb
phrase. In that case, removing them would likely loose the meaning of the sen-
tence. PPs attached to VPs that are positioned after the head verb are therefore
not pruned. However, PPs attached to VPs that are positioned prior to the verb
are considered less likely to be mandatory and are removed. Removing PPs based
solely on syntactic information will likely make mistakes. PPs that do not con-
tain necessary information may be kept, and vice-versa. However, the purpose of
this heuristic is to prune as cautiously as possible. Sections 3.2 and 3.3 describe
heuristics that take semantics into account.

3.2   Syntax and Relevancy Based Pruning

The danger with syntax only based pruning is that it may remove sub-trees that
do contain relevant information for the summary. In order to avoid this, we toned
down our syntax based heuristics described in Section 3.1, by measuring the rele-
vancy of the sub-tree to prune and only remove it if it is below a certain threshold.
In the case of query-based summarization, we used the cosine similarity between
the tf-idf values of the candidate sub-tree to prune and the topic/query. Specifi-
cally, if the syntax based heuristic consider that a sub-tree is a candidate for prun-
ing but its similarity with the topic & query is above some threshold, we do not
prune it on the grounds that it seems to have relevant content.

3.3   Relevancy-Driven Syntactic Pruning

Our previous techniques (see Sections 3.1 and 3.2) focused on keeping the sentence
grammaticality as much as possible by driving the pruning based on syntax. Next
we took an approach to prune sentences focusing less on preserving grammatical-
ity and more on preserving relevant information. Our last approach focused on
finding irrelevant information within a sentence and remove its embedding sub-
tree. Specifically, we parse the extracted sentences as before and for each sub-tree

except for noun phrases, verb phrases or individual words, we calculate its cosine similarity with the topic/query based on tf-ifd values. Sub-trees below a certain threshold are pruned; the others are kept. We do not allow pruning of noun phrases, verb phrases and individual words in order to preserve a minimal grammaticality; all other phrase types, are however possible candidates for pruning. For example, consider the following scenario:

Topic: *Turkey and the European Union*

Query: *What positive and negative developments have there been in Turkey's efforts to become a formal member of the European Union?*

(8) *Turkey had been asking for three decades to join the European Union but its demand was turned away by the European Union in December 1997 that led to a deterioration of bilateral relations.*

Here, Sentence 8 is the original candidate extracted from the corpus. Its parse tree generated by the Stanford Parser [22] is shown is Figure 1, with the relevancy score indicated in bold. For example, the sub-tree rooted by the SBAR *(that led to a deterioration of bilateral relations)* was computed to have a relevance of 0.0 with the topic and the query. All sub-trees rooted at a node whose relevance is smaller than some threshold value are pruned. If we set $t = 0$ (i.e. any relevance with topic/query will be considered useful), the above sentence would therefore be compressed as:
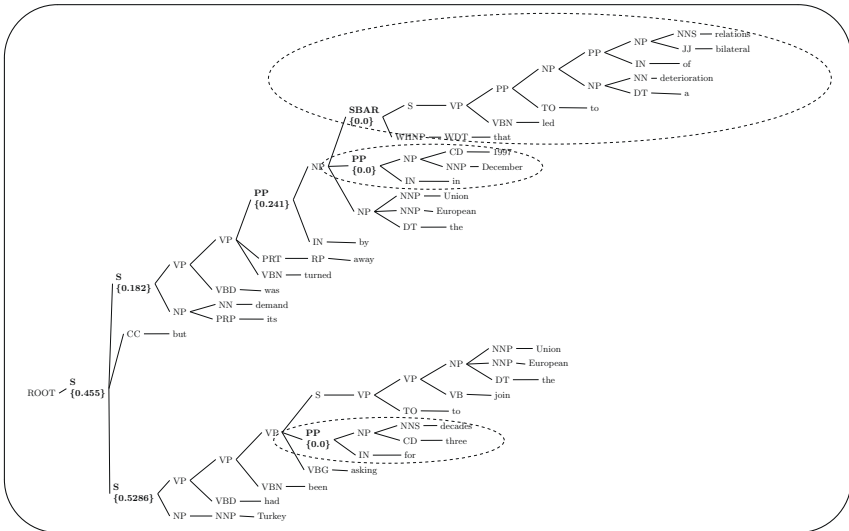
Fig. 1. Dependency Phrase Structure For Sentence 8

(8c) *Turkey had been asking* ~~for three decades~~ *to join the European Union but its demand was turned away by the European Union* ~~in December 1997 that led to a deterioration of bilateral relations.~~

4   Evaluation

To evaluate our pruning techniques extrinsically for the purpose of summary generation, we used two standard text corpora available for summarization: the Text Analysis Conference (TAC) 2008 [23], which provides a text corpus created from blogs and the Document Understanding Conference (DUC) 2007 [18] which provides a text corpus of news articles. To ensure that our results were not tailored to one specific summarizer, we used two different systems: BlogSum [24], an automatic summarizer based on discourse relations and MEAD [25], a generalized automatic summarization system. In order to generate syntactic trees for our experiment, we used the Stanford Parser [22]. To evaluate each compression technique, we generated summaries without any compression and compared the results based on two metrics: compression rates and ROUGE scores for content evaluation.

4.1   Evaluation of Compression Rates

To measure the compression rate of each technique, we first created summaries using BlogSum and MEAD, setting a limit of 250 words per summary, then applied each sentence pruning heuristic independently to generate different sets of summaries.

Syntax-Driven Pruning. Table 1 shows the compression rates achieved by each heuristic for both summarizers and both datasets. As Table 1 shows, with both datasets, apart from the combined approach, the highest sentence compression was achieved by preposition based pruning (PP pruning); while the lowest compressions were observed with relative clause (RC), adverbial phrases (Adv) and trailing conjuncted verb phrases (TC-VP) pruning. This is not surprising as PPs are a priori more frequent than the other syntactic constructions. Also not surprisingly, the combined approach which applies all pruning heuristics achieved the highest compression rate in both datasets reaching about 73% to 75% compression rates.

Syntax and Relevancy Based Pruning. Table 2 shows the compression rate achieved by each heuristic using the syntax and relevancy based pruning. As the results show, with both datasets, the compression effect of each heuristic has been toned down, but the relative ranking of the heuristics are the same. This seems to imply that each type of syntactic phrase is as likely to contain irrelevant information; and one particular construction should not be privileged for pruning

Table 1. Sentence Compression Rates of Syntax-Driven Pruning

| | BlogSum | | | | MEAD | | | |
| | TAC 2008 | | DUC 2007 | | TAC 2008 | | DUC 2007 | |
	No of. Words	Compression Rate	No of. Words	Compression Rate	No of. Words	Compression Rate	No of. Wordss	Compression Rate
Original	11272	100.0%	10648	100.0%	11759	100.0%	11186	100.0%
Adv Pruning	10804	95.8%	10422	97.9%	11491	97.7%	10973	98.1%
RC Pruning	10803	95.8%	10309	96.8%	11273	95.9%	10708	95.7%
TC-VP Pruning	10887	96.6%	10271	96.5%	11530	98.0%	10789	96.4%
Adj Pruning	10430	92.5%	9897	93.0%	11225	95.4%	10391	92.9%
PP Pruning	9349	83.0%	8442	79.3%	10359	76.7%	8584	76.3%
Combined	8170	72.5%	7995	75.1%	9799	83.3%	8143	72.8%

purposes. Overall, when all pruning heuristics are combined, the relevancy factor reduces the pruning by about 8 to 11% (from 73-75% to 82-86%) .[2]

Table 2. Sentence Compression Rates of Syntax and Relevancy Based Pruning

| | BlogSum | | | | MEAD | | | |
| | TAC 2008 | | DUC 2007 | | TAC 2008 | | DUC 2007 | |
	No of. Words	Compression Rate	No of. Words	Compression Rate	No of. Words	Compression Rate	No of. Wordss	Compression Rate
Original	11272	100.0%	10648	100.0%	11759	100.0%	11186	100.0%
Adv Pruning	10869	96.4%	10435	98.0%	11526	98.0%	11030	98.6%
RC Pruning	11100	98.4%	10575	99.3%	11495	97.7%	10988	98.2%
TC-VP Pruning	10887	96.6%	10478	98.4%	11644	99.0%	10969	98.1%
Adj Pruning	11111	98.6%	10085	94.7%	11287	96.0%	10535	94.2%
PP Pruning	10261	91.0%	9834	92.3%	10976	93.3%	9754	87.2%
Combined	9234	82.0%	9170	86.1%	10361	88.1%	9178	82.0%

Relevancy-Driven Syntactic Pruning. Table 3 shows the results of the compression rate achieved by relevancy-driven syntactic pruning. The relevancy-driven syntactic pruning has achieved a higher compression rate than syntax and relevancy based pruning. Table 4 shows the types of syntactic structures that were

Table 3. Sentence Compression Rates of Relevancy-Driven Syntactic Pruning

| | BlogSum | | | | MEAD | | | |
| | TAC 2008 | | DUC 2007 | | TAC 2008 | | DUC 2007 | |
	No Of. Words	Compression Rate	No Of. Words	Compression Rate	No Of. Words	Compression Rate	No Of. Words	Compression Rate
Original	11272	100.0%	10648	100.0%	11759	100.0%	11186	100.0%
Relevancy-Driven	7457	66.1%	7879	74.0%	7122	60.6%	6801	69.0%

removed by the relevancy-driven pruning and their relative frequencies. As the results shows, the most frequent syntactic structures removed were PPs and the least were adverbial phrases (Adv). This result correlates with our syntax-driven pruning as we achieved similar individual compression rates for these phrase structures.

---

[2] The reduction rate is of course proportional to the relevancy threshold used (see Section 3.2). In this experiment, we set the threshold to be the most conservative (t = 0), hence keeping everything that has any relevance to the topic/query.

Table 4. Syntactic Phrase Structures Removed by Relevancy-Driven Pruning

	BlogSum				MEAD			
	TAC 2008		DUC 2007		TAC 2008		DUC 2007	
	No of. Phrases	Relative Frequency	No of. Phrases	Relative Frequency	No of. Phrases	Relative Frequency	No of. Phrases	Relative Frequency
PP Pruning	395	50.5%	402	62.4%	177	42.3%	408	63.6%
Other	189	24.1%	136	29.3%	157	31.6%	149	30.1%
RC Pruning	94	12.0%	56	8.7%	44	10.5%	59	9.2%
Adj Pruning	75	9%	35	5.4%	26	6.2%	20	3.1%
Adv Pruning	29	3.7%	15	2.4%	14	3.3%	5	1.0%
Total	782	100%	644	100%	418	100%	641	100%

4.2   Evaluation of Content

Compression rate is interesting, but not at the cost of pruning useful information. In order to measure the effect of the pruning strategies on the content of summaries, we have ran the same experiments again but this time we have calculated the F-measures of the ROUGE scores (R-2 and R-SU4). In principle, pruning sentences should shorten summaries thus allowing us to fill the summary with new relevant sentences and hence improve its overall content. In order to evaluate the effect of sentence compression on this, we first created summaries with a word limit of 250 and then created two summaries: one without filling the summary with extra content to reach the 250 word limit and one with filling with new content to reach the 250 word limit. We calculated ROUGE scores for both set of summaries. Again to avoid any bias, we created summaries using both the Blog-Sum and MEAD systems and both datasets.

Syntax-Driven Pruning. Tables 5 and 6 show the results obtained with and without content filling respectively. Table 5 show a drop in ROUGE score for both summarization systems and both datasets. This goes against our hypothesis that by default specific syntactic constructions can be removed without losing much content. In addition, when filling the summary with extra sentences, ROUGE scores do seem to improve (as shown in Table 6); however Pearson's χ^2 and t-tests show that this difference is not statistically significant. What is more surprising is that this phenomenon is true for the combined heuristics, but also for each individual pruning heuristic.

Table 5. Content Evaluation of Compressed Summaries with Syntax-Driven Pruning (Without Filling)

	BlogSum				MEAD			
	TAC 2008		DUC 2007		TAC 2008		DUC 2007	
	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4
Original	0.074	0.112	0.088	0.141	0.040	0.063	0.086	0.139
Adv Pruning	0.074	0.113	0.089	0.143	0.039	0.063	0.086	0.139
RC Pruning	0.072	0.109	0.087	0.140	0.039	0.062	0.085	0.138
TC-VP Pruning	0.073	0.111	0.088	0.140	0.040	0.063	0.085	0.137
Adj Pruning	0.068	0.108	0.084	0.140	0.038	0.063	0.080	0.136
PP Pruning	0.065	0.103	0.072	0.121	0.035	0.056	0.069	0.117
Combined	0.060	0.100	0.074	0.128	0.034	0.056	0.068	0.121

Table 6. Content Evaluation of Compressed Summaries with Syntax-Driven Pruning (With Filling)

	BlogSum				MEAD			
	TAC 2008		DUC 2007		TAC 2008		DUC 2007	
	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4
Original	0.074	0.112	0.088	0.141	0.040	0.063	0.086	0.140
Adv Pruning	0.075	0.114	0.090	0.143	0.044	0.063	0.087	0.140
RC Pruning	0.073	0.111	0.088	0.141	0.039	0.062	0.086	0.140
TC-VP Pruning	0.073	0.111	0.089	0.141	0.040	0.062	0.086	0.139
Adj Pruning	0.075	0.110	0.085	0.142	0.038	0.063	0.082	0.140
PP Pruning	0.070	0.131	0.079	0.131	0.035	0.058	0.076	0.127
Combined	0.065	0.139	0.065	0.139	0.035	0.060	0.077	0.135

Table 7. Content Evaluation of Compressed Summaries with Syntax-Driven with Relevancy Pruning (Without Filling)

	BlogSum				MEAD			
	TAC 2008		DUC 2007		TAC 2008		DUC 2007	
	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4
Original	0.074	0.112	0.088	0.141	0.040	0.063	0.086	0.139
Adv Pruning	0.074	0.113	0.089	0.143	0.039	0.063	0.087	0.140
RC Pruning	0.073	0.110	0.088	0.141	0.039	0.062	0.086	0.139
TC-VP Pruning	0.073	0.111	0.088	0.141	0.040	0.063	0.085	0.138
Adj Pruning	0.070	0.110	0.086	0.142	0.038	0.063	0.082	0.138
PP Pruning	0.072	0.110	0.086	0.137	0.039	0.062	0.079	0.129
Combined	0.069	0.110	0.085	0.140	0.038	0.061	0.078	0.132

Syntax and Relevancy Based Pruning. Recall that the syntax-driven pruning did not consider the relevancy of the sub-tree to prune. When we do take the relevancy to account; surprisingly the ROUGE scores do not improve significantly either. Tables 7 and 8 show the ROUGE scores of the compressed summaries based on syntax and relevancy without filling (Table 7) and with content filling (Table 8). Again any semblance of improvement is not statistically significant.

Relevancy-Driven Pruning. Table 9 shows the results of relevancy-driven pruning with and without filling and compares them to the original summaries. Again the results are surprisingly low. This last approach was still not able to improve ROUGE scores significantly.

4.3   Discussion

Although the results of the compression rate were inline with previous work [7,20], we were surprised at the results of the content evaluation. However this might explain why, to our knowledge, so little work can be found in the literature on the evaluation of syntactic sentence pruning for summarization. Our pruning heuristics could of course be fine-tuned to be more discriminating. We could, for example, use verb frames or lexical-grammatical rules to prune PPs; but we do not foresee a significant increase in ROUGE scores. The relevance measure that we

Table 8. Content Evaluation of Compressed Summaries with Syntax-Driven with Relevancy Pruning (With Filling)

	BlogSum				MEAD			
	TAC 2008		DUC 2007		TAC 2008		DUC 2007	
	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4
Original	0.074	0.112	0.088	0.141	0.040	0.063	0.086	0.140
Adv Pruning	0.075	0.114	0.090	0.143	0.040	0.063	0.087	0.140
RC Pruning	0.072	0.111	0.088	0.141	0.040	0.062	0.086	0.140
TC-VP Pruning	0.074	0.111	0.089	0.141	0.040	0.062	0.086	0.139
Adj Pruning	0.072	0.111	0.086	0.142	0.038	0.063	0.085	0.141
PP Pruning	0.072	0.111	0.088	0.141	0.040	0.062	0.084	0.135
Combined	0.071	0.112	0.088	0.145	0.037	0.062	0.085	0.141

Table 9. Content Evaluation of Compressed Summaries with Relevancy-Driven Syntactic Pruning (With and Without Filling)

	BlogSum				MEAD			
	TAC 2008		DUC 2007		TAC 2008		DUC 2007	
	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4
Original	0.074	0.112	0.088	0.141	0.040	0.063	0.086	0.139
Relevancy-Driven Without Filling	0.065	0.100	0.077	0.125	0.034	0.055	0.066	0.110
Relevancy-Driven With Filling	0.068	0.106	0.083	0.135	0.033	0.060	0.078	0.128

used (see Section 3.3) could also be experimented with, but again, we do not expect much increase from that end. Using a better performing summarizer might also be a possible avenue of investigation to provide us with better input sentences and better "filling" sentences after compression.

5   Conclusion and Future Work

In this paper, we have described our experiments on syntactic based sentence pruning applied to automatic text summarization. We have defined three types of pruning techniques based on complete syntactic parses: a first technique based solely on syntax, a second technique that tones down the syntactic pruning by taking relevancy into account and third technique that is driven by relevancy. These techniques were applied to the sentences extracted by two different summarizers to generate compressed summaries and evaluated on the TAC-2008 and DUC-2007 benchmarks. According to results, these pruning techniques generate a compression rate between 60% to 88% which is inline with previous work [7,20]. However, when using the extra space to include additional sentences, the content evaluation does not show a significant improvement in ROUGE scores.

As future work, we are planning to move to a manual human evaluation, as [3] and [11] did in their work. We are interested to find out if human assesssors agree with ROUGE scores, and thus we need to re-think our syntactic approach or if a human evaluation does consider the condensed summaries to be more informative than the original ones, hence putting aside ROUGE measures for the task.

References

1. Chandrasekar, R., Doran, C., Srinivas, B.: Motivations and Methods for Text Simplification. In: Proceedings of COLING 1996, Copenhagen, pp. 1041–1044 (1996)
2. Dorr, B., Zajic, D., Schwartz, R.: Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation. In: Proceedings of the HLT-NAACL Workshop on Text Summarization, pp. 1–8 (2003)
3. Knight, K., Marcu, D.: Summarization beyond sentence extraction: A probabilistic approach to sentence compression. Artificial Intelligence 139(1), 91–107 (2002)
4. Hahn, U., Mani, I.: The Challenges of Automatic Summarization. IEEE Computer
5. Murray, G., Joty, S., Ng, R.: The University of British Columbia at TAC 2008. In: Proceedings of TAC 2008, Gaithersburg, Maryland, USA (2008)
6. Jing, H.: Sentence Reduction for Automatic Text Summarization. In: Proceedings of the Sixth Conference on Applied Natural Language Processing, Seattle, pp. 310–315 (April 2000)
7. Gagnon, M., Da Sylva, L.: Text Compression by Syntactic Pruning. In: Lamontagne, L., Marchand, M. (eds.) Canadian AI 2006. LNCS (LNAI), vol. 4013, pp. 312–323. Springer, Heidelberg (2006)
8. Jaoua, M., Jaoua, F., Belguith, L.H., Hamadou, A.B.: Évaluation de l'impact de l'intégration des étapes de filtrage et de compression dans le processus d'automatisation du résumé. In: Résumé Automatique de Documents. Document numérique, Lavoisier, vol. 15, pp. 67–90 (2012)
9. Jing, H., McKeown, K.R.: Cut and Paste Based Text Summarization. In: Proceedings of NAACL-2000, Seattle, pp. 178–185 (2000)
10. Conroy, J.M., Schlesinger, J.D., O'Leary, D.P., Goldstein, J.: Back to Basics: CLASSY 2006. In: Proceedings of the HLT-NAACL 2006 Document Understanding Workshop, New York City (2006)
11. Nguyen, M.L., Phan, X.H., Horiguchi, S., Shimazu, A.: A New Sentence Reduction Technique Based on a Decision Tree Model. International Journal on Artificial Intelligence Tools 16(1), 129–138 (2007)
12. McClosky, D., Charniak, E., Johnson, M.: Effective Self-Training for Parsing. In: Proceedings of HLT-NAACL 2006, New York, pp. 152–159 (2006)
13. Fellbaum, C.: WordNet: An Electronic Lexical Database. The MIT Press (May 1998)
14. Le Nguyen, M., Shimazu, A., Horiguchi, S., Ho, B.T., Fukushi, M.: Probabilistic Sentence Reduction Using Support Vector Machines. In: Proceedings of COLING 2004, Geneva, pp. 743–749 (August 2004)
15. Clarke, J., Lapata, M.: Global Inference for Sentence Compression an Integer Linear Programming Approach. Journal of Artificial Intelligence Research (JAIR) 31(1), 399–429 (2008)
16. Filippova, K., Strube, M.: Dependency Tree Based Sentence Compression. In: Proceedings of the Fifth International Natural Language Generation Conference, INLG 2008, Stroudsburg, PA, USA, pp. 25–32 (2008)
17. Schlesinger, J.D., O'Leary, D.P., Conroy, J.M.: Arabic/English Multi-document Summarization with CLASSY: The Past and the Future. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 568–581. Springer, Heidelberg (2008)

18. Dang, H.T.: DUC 2005: Evaluation of Question-focused Summarization Systems. In: Proceedings of the Workshop on Task-Focused Summarization and Question Answering, Sydney, pp. 48–55 (2006)
19. Dang, H.T.: Overview of DUC 2006. In: Proceedings of the HLT-NAACL 2006 Document Understanding Workshop (2006)
20. Zajic, D.M., Dorrand, B.J., Lin, J., Schwartz, R.: Multi-candidate Reduction: Sentence Compression as a Tool for Document Summarization Tasks. Information Processing and Management 43(6), 1549–1570 (2007)
21. Harman, D., Liberman, M.: TIPSTER Complete. Linguistic Data Consortium (LDC), Philadelphia (1993)
22. Marneffe, M.C.D., Manning, C.D.: The Stanford typed dependencies representation. In: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, CrossParser 2008, Manchester, pp. 1–8 (2008)
23. Dang, H., Owczarzak, K.: Overview of the TAC 2008 Update Summarization Task. In: Proceedings of the Text Analysis Conference, TAC 2008, Gaithersburg (2008)
24. Mithun, S.: Exploiting Rhetorical Relations in Blog Summarization. In: Farzindar, A., Kešelj, V. (eds.) Canadian AI 2010. LNCS, vol. 6085, pp. 388–392. Springer, Heidelberg (2010)
25. Radev, D., Allison, T., Blair-Goldensohn, S., Blitzer, J., Çelebi, A., Dimitrov, S., Drabek, E., Hakim, A., Lam, W., Liu, D., Otterbacher, J., Qi, H., Saggion, H., Teufel, S., Topper, M., Winkel, A., Zhang, Z.: MEAD - A platform for multidocument multilingual text summarization. In: Proceedings of LREC 2004, Lisbon, Portugal (May 2004)

An Unsupervised Aspect Detection Model
for Sentiment Analysis of Reviews

Ayoub Bagheri[1], Mohamad Saraee[2], and Franciska de Jong[3]

[1] Intelligent Database, Data Mining and Bioinformatics Lab, Electrical and Computer Engineering Department, Isfahan University of Technology, Isfahan, Iran
a.bagheri@ec.iut.ac.ir
[2] School of Computing, Science and Engineering, University of Salford, Manchester, UK
m.saraee@salford.ac.uk
[3] University of Twente, Human Media Interaction, P.O. Box 217, 7500 AE Enschede, The Netherlands
f.m.g.dejong@utwente.nl

Abstract. With the rapid growth of user-generated content on the internet, sentiment analysis of online reviews has become a hot research topic recently, but due to variety and wide range of products and services, the supervised and domain-specific models are often not practical. As the number of reviews expands, it is essential to develop an efficient sentiment analysis model that is capable of extracting product aspects and determining the sentiments for aspects. In this paper, we propose an unsupervised model for detecting aspects in reviews. In this model, first a generalized method is proposed to learn multi-word aspects. Second, a set of heuristic rules is employed to take into account the influence of an opinion word on detecting the aspect. Third a new metric based on mutual information and aspect frequency is proposed to score aspects with a new bootstrapping iterative algorithm. The presented bootstrapping algorithm works with an unsupervised seed set. Finally two pruning methods based on the relations between aspects in reviews are presented to remove incorrect aspects. The proposed model does not require labeled training data and can be applicable to other languages or domains. We demonstrate the effectiveness of our model on a collection of product reviews dataset, where it outperforms other techniques.

Keywords: sentiment analysis, opinion mining, aspect detection, review mining.

1 Introduction

In the past few years, with the rapid growth of user-generated content on the internet, sentiment analysis (or opinion mining) has attracted a great deal of attention from researchers of data mining and natural language processing. Sentiment analysis is a type of text analysis under the broad area of text mining and computational intelligence. Three fundamental problems in sentiment analysis are: aspect detection, opinion word detection and sentiment orientation identification [1-2].

Aspects are topics on which opinion are expressed. In the field of sentiment analysis, other names for aspect are: features, product features or opinion targets [1-5].

Aspects are important because without knowing them, the opinions expressed in a sentence or a review are of limited use. For example, in the review sentence "after using it, I found the size to be perfect for carrying in a pocket", "size" is the aspect for which an opinion is expressed. Likewise aspect detection is critical to sentiment analysis, because its effectiveness dramatically affects the performance of opinion word detection and sentiment orientation identification. Therefore, in this study we concentrate on aspect detection for sentiment analysis.

Existing aspect detection methods can broadly be classified into two major approaches: supervised and unsupervised. Supervised aspect detection approaches require a set of pre-labeled training data. Although the supervised approaches can achieve reasonable effectiveness, building sufficient labeled data is often expensive and needs much human labor. Since unlabeled data are generally publicly available, it is desirable to develop a model that works with unlabeled data. Additionally due to variety and wide range of products and services being reviewed on the internet, supervised, domain-specific or language-dependent models are often not practical. Therefore the framework for the aspect detection must be robust and easily transferable between domains or languages.

In this paper, we present an unsupervised model which addresses the core tasks necessary to detect aspects from review sentences in a sentiment analysis system. In the proposed model we use a novel bootstrapping algorithm which needs an initial seed set of aspects. Our model requires no labeled training data or additional information, not even for the seed set. The model can easily be transform between domains or languages. In the remainder of this paper, detailed discussions of existing works on aspect detection will be given in section 2.Section 3 describes the proposed aspect detection model for sentiment analysis, including the overall process and specific designs. Subsequently we describe our empirical evaluation and discuss important experimental results in section 4. Finally we conclude with a summary and some future research directions in section 5.

2   Related Work

Several methods have been proposed, mainly in the context of product review mining [1-14]. The earliest attempt on aspect detection was based on the classic information extraction approach of using frequently occurring noun phrases presented by Hu and Liu [3]. Their work can be considered as the initiator work on aspect extraction from reviews. They use association rule mining (ARM) based on the Apriori algorithm to extract frequent itemsets as explicit product features, only in the form of noun phrases. Their approach works well in detecting aspects that are strongly associated with a single noun, but are less useful when aspects encompass many low-frequency terms. The proposed model in our study works well with low-frequency terms and uses more POS patterns to extract the candidates for aspect. Wei et al. [4] proposed a semantic-based product aspect extraction (SPE) method. Their approach begins with preprocessing task, and then employs the association rule mining to identify candidate product aspects. Afterward, on the basis of the list of positive and negative opinion

words, the semantic-based refinement step identifies and then removes from the set of frequent aspects possible non-product aspects and opinion-irrelevant product aspects. The SPE approach relies primarily on frequency- and semantic-based extraction for the aspect detection, but in our study we use frequency-based and inter-connection information between the aspects and give more importance to multi-word aspects and aspects with an opinion word in the review sentence. Somprasertsri and Lalitroj-wong's [8] proposed a supervised model for aspect detection by combining lexical and syntactic features with a maximum entropy technique. They extracted the learning features from an annotated corpus. Their approach uses a maximum entropy classifier for extracting aspects and includes the postprocessing step to discover the remaining aspects in the reviews by matching the list of extracted aspects against each word in the reviews. We use Somprasertsri and Lalitrojwong's work for a comparison to our proposed model, because the model in our study is completely unsupervised.

Our work on aspect detection designed to be as unsupervised as possible, so as to make it transferable through different types of domains, as well as across languages. The motivation is to build a model to work on the characteristics of the words in reviews and interrelation information between them, and to take into account the influence of an opinion word on detecting the aspect.

3   Aspect Detection Model for Sentiment Analysis

Figure 1 gives an overview of the proposed model used for detecting aspects in sentiment analysis. Below, we discuss each of the functions in aspect detection model in turn.

Model: Aspect Detection for Sentiment Analysis
Input: Reviews Dataset
Method:

       Extract Review Sentences
       FOR each sentence
            Use POS Tagging
            Extract POS Tag Patterns as Candidates for Aspects
       END FOR
       FOR each candidate aspect
            Use Stemming
            Select Multi-Word Aspects
            Use a Set of Heuristic Rules
       END FOR
       Make Initial Seeds for Final Aspects
       Use Iterative Bootstrapping for Detecting Final Aspects
       Aspect Pruning
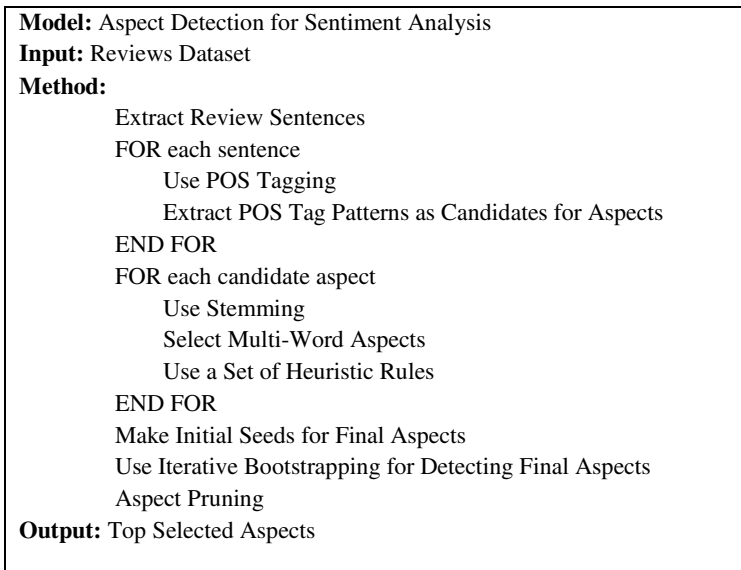**Output:** Top Selected Aspects

---

Fig. 1. The proposed model for aspect detection for sentiment analysis

3.1   Candidate Generation

In this paper we focus on five POS (Part-Of-Speech) tags: NN, JJ, DT, NNS and VBG, where they are the tags for nouns, adjectives, determiners, plural nouns and verb gerunds respectively [15]. Additionally stemming is used to select one single form of a word instead of different forms [16]. Based on the observation that aspects are nouns, we extract combination of noun phrases and adjectives from review sentences. We use several POS patterns introduced in table 1.

Table 1. Heuristic combination POS patterns for candidate generation

Description	Patterns
Combination of nouns	Unigram to four-gram of NN and NNS
Combination of nouns and adjectives	Bigram to four-gram of JJ, NN and NNS
Combination determiners and adjectives	Bigram of DT and JJ
Combination of nouns and verb gerunds (present participle)	Bigram to trigram of DT, NN, NNS and VBG

3.2   Multi-Word Aspects

In the review sentences, some aspects that people talk about have more than one single word, "battery life", "signal quality" and "battery charging system" are examples. This step is to find useful multi-word aspects from the reviews. A multi-word aspect is represented by $a = a_1 a_2 \dots a_n$ where a_i represents a single-word contained in a, and n is the number of words contained in a. In this paper, we propose a generalized version of FLR method [17, 18] to rank the extracted multi-word aspects and select the importance ones. FLR is a word scoring method that uses internal structures and frequencies of candidates. The FLR for an aspect a is calculated as:

$$FLR(a) = f(a) * LR(a) \tag{1}$$

Where $f(a)$ is the frequency of aspect a, or in the other words it is number of the sentences in the corpus which a is appeared, and $LR(a)$ is LR score of aspect a which is defined as a geometric mean of the scores of subset single words as:

$$LR(a) = (lr(a_1) * lr(a_2) * \dots * lr(a_n))^{\frac{1}{n}} \tag{2}$$

The left score $l(a_i)$ of each word a_i of a target aspect is defined as the number of types of words appearing to the left of a_i, and the right score $r(a_i)$ is defined in the same manner. An LR score for single word a_i is defined as:

$$lr(a_i) = \sqrt{l(a_i) * r(a_i)} \tag{3}$$

The proposed generalization of the FLR method is on the definition of two parameters: $l(a_i)$ and $r(a_i)$. We change the definitions to give more importance to the

aspects with more containing words. In the new definition, in addition to the frequency we consider position of a_i in aspect a. For the score $l(a_i)$ of each word a_i of a target aspect, we not only consider a single word on the left of a_i, but we consider if there is more than one word on the left. We assign a weight for each position, that this weight is equal to one for the first word on the left, is two for the second word and so on. We define the score $r(a_i)$ in the same manner. In addition, we apply the add-one smoothing to both of them to avoid the score being zero when a_i has no connected words.

3.3   Heuristic Rules

With finding the candidates, we need to move to the next level, aspect identification. For this matter we start with heuristic and experimentally extracted rules. Below, we present Rule #1 and Rule #2 for the aspect detection model.

Rule #1: Remove aspects which there are no opinion words with in a sentence.

Rule #2: Remove aspects that contain stop words.

3.4   Unsupervised Initial Seed Set

In this function we focus on selecting some aspects from the candidates as seed set information. We introduce a new metric named A-Score, which selects the seed set in an unsupervised manner. This metric is employed to learn a small list of top aspects with a complete precision.

3.5   A-Score Metric

Here we introduce a new metric, named A-score which uses inter-relation information between words to score them. We score each candidate aspect with A-score metric defined as:

$$A - Score(a) = f(a) * \sum_i \log_2 \left(\frac{f(a,b_i)}{f(a)*f(b_i)} * N + 1 \right) \tag{4}$$

Where a is the current aspect, $f(a)$ is the number of the sentences in the corpus which a is appeared, $f(a, b_i)$ is the frequency of co-occurrence of aspect a and b_i in each sentence. b_i is ith aspect in the list of seed aspects, and N is number of sentences in the corpus. The A-Score metric is based on mutual information between an aspect and a list of aspects, in addition it considers frequency of each aspect. We apply the add-one smoothing to the metric, so all co-frequencies be non-zero. This metric helps to extract more informative aspects and more co-related ones.

3.6   Iterative Bootstrapping Algorithm for Detecting Aspects

Iterative bootstrapping algorithm focuses on to learn final list of aspects from a small number of unsupervised seed set information. Bootstrapping can be viewed as an

iterative clustering technique which in each iteration, the most interesting and valuable candidate is chosen to adjust the current seed set. This technique continues until satisfying a stopping criterion like a predefined number of outputs. The important part in an iterative bootstrapping algorithm is how to measure the value score of each candidate in each iteration. The proposed iterative bootstrapping algorithm for detecting aspects is shown in figure 2. In this algorithm we use A-score metric to measure the value score of each candidate in each iteration.

Algorithm: Iterative Bootstrapping for Detecting Aspects
Input: Seed Aspects, Candidate Aspects
Method:
       FOR each candidate aspect
          Calculate A-Score
          Add the Aspect with Maximum A-Score to the Seed Aspects
       END FOR
       Copy Seed Aspects to Final Aspects
Output: Final Aspects

Fig. 2. The proposed iterative bootstrapping algorithm for detecting aspects

From figure 2, the task of the proposed iterative bootstrapping algorithm is to enlarge the initial seed set into a final list of aspects. In each iteration, the current version of the seed set and the list of candidate aspects are used to find the value score of A-Score metric for each candidate, resulting one more aspect for the seed set. Finally, the augmented seed set is the final aspect list and the output of the algorithm.

3.7 Aspect Pruning

After finalizing the list of aspects, there may exist redundant selected ones. For instances, "Suite" or "Free Speakerphone" are both redundant aspects, while "PC Suite" and "Speakerphone" are meaningful ones. Aspect pruning aims to remove these kinds of redundant aspects. For aspect pruning, we propose two kinds of pruning methods: Subset-Support Pruning and Superset-Support Pruning. We extracted these methods based on the experiment studies in our research.

Subset-Support Pruning
As we can see from table 1, two of the POS patterns are "JJ NN" and "JJ NN NN". These patterns extract some useful and important aspects like "remote control" or "optical zoom", but there are some redundant and meaningless aspects regarding to these patterns. Aspects like "free speakerphone" or "rental dvd player" are examples, while subset of them "speakerphone" or "dvd player" are useful aspects. This step checks multi-word aspects that start with an adjective (JJ POS pattern), and removes

those that are likely to be meaningless. In this step we remove the adjective part for aspects and then check a threshold if the second part is meaningful.

Superset-Support Pruning

In this step we remove redundant single word aspects. We filter single-word aspects which there is a superset ones of them. "Suite" or "life" are both examples of these redundant aspects which "PC Suite" or "battery life" are superset meaningful ones.

4   Experimental Results

In this section we discuss the experimental results for the proposed model and presented algorithms. We employed datasets of customer reviews for five products for our evaluation purpose (available at http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#datasets). This dataset focus on electronic products: Apex AD2600 Progressive-scan DVD player, Canon G3, Creative Labs Nomad Jukebox Zen Xtra 40 GB, Nikon Coolpix 4300, and Nokia 6610. Table 2 shows the number of manually tagged product aspects and the number of reviews for each product in the dataset.

Table 2. Summary of customer review dataset

Dataset	Number of reviews	No. of manual aspects
Canon	45	100
Nikon	34	74
Nokia	41	109
Creative	95	180
Apex	99	110

4.1   Comparative Study

In our evaluation, after preprocessing and extracting the candidates, we score each multi-word aspect with the generalized FLR method and select those with the score higher than the average, and then we merge single-word and multi-word aspects in a list. Heuristic rules are then employed for the whole list of single and multi-word aspects to take into account the influence of an opinion word on detecting the aspect and remove useless aspects.

Finding an appropriate number of good seeds for bootstrapping algorithm is an important step. In our experiments we used A-score metric to extract automatically the seed set. We have experimented with different numbers of seeds (i.e., 5, 10, 15 and 20) for iterative bootstrapping, and found that the best number of the seeds is about 10 to 15. Therefore seeds were automatically chosen for iterative bootstrapping

algorithm, and the stopping criterion is defined when about 70 to 120 aspects have been learned. For the subset-support pruning method we set the threshold 0.5. In su-perset-support pruning step if an aspect has a frequency lower than three and its ratio to the superset aspect is less than experimentally threshold set one, it is pruned. Table 3 shows the experimental results of our model at three main steps described in section 3, Multi-word aspects and heuristic rules, Iterative bootstrapping with A-Score and Aspect pruning steps.

Table 2. Recall and precision at three main steps of the proposed model

Dataset	Multi-word aspects and heuristic rules	Iterative boot-strapping with A-Score	Aspect pruning
Precision			
Canon	26.7	75.0	83.1
Nikon	28.4	69.8	87.5
Nokia	23.9	73.5	79.0
Creative	14.8	79.2	88.9
Apex	19.3	78.8	82.0
Recall			
Canon	85.7	74.0	70.1
Nikon	82.4	72.5	68.6
Nokia	84.1	72.5	71.0
Creative	78.9	59.2	56.3
Apex	74.6	65.1	65.1

Table 3 gives all the precision and recall results at the main steps of the proposed model. In this table, column 1 lists each product. Each column gives the precision and recall for each product. Column 2 uses extracted single-word aspects and selected multi-word aspects based on generalized FLR approach and employing heuristic rules for each product. The results indicate that extracted aspects contain a lot of errors. Using this step alone gives poor results in precision. Column 3 shows the corresponding results after employing Iterative bootstrapping algorithm with A-Score metric. We can see that the precision is improved significantly by this step but the recall drops. Column 4 gives the results after pruning methods are performed. The results demonstrate the effectiveness of the pruning methods. The precision is improved dramatically, but the recall drops a few percent.

We evaluate the effectiveness of the proposed model compared with the benchmarked results by [4]. Wei et al. proposed a semantic-based product aspect extraction (SPE) method and compared the results of the SPE with the association rule mining approach (ARM) given in [3]. The SPE technique exploits a list of positive and negative adjectives defined in the General Inquirer to recognize opinion words semantically and subsequently extract product aspects expressed in customer reviews.

Table 3. Experiment results of comparative study

Product	ARM		SPE		Proposed model	
	Precision	Recall	Precision	Recall	Precision	Recall
Canon	51.1	63.0	48.7	75.0	83.1	70.1
Nikon	51.0	67.6	47.4	75.7	87.5	68.6
Nokia	49.5	57.8	56.5	72.5	79.0	71.0
Creative	37.0	56.1	44.0	65.0	88.9	56.3
Apex	51.0	60.0	52.4	70.0	82.0	65.1
Macro avg.	47.9	60.9	49.8	71.6	84.1	66.2
Micro avg.	46.1	59.9	48.6	70.5	83.6	66.2

Table 4 shows the experimental results of our model in comparison with SPE and ARM techniques (the values in this Table for ARM and SPE come from the results in [4]). Both the ARM and SPE techniques employ a minimum support threshold set at 1% in the frequent aspect identification step for finding aspects according to the association rule mining.

From Table 4, the macro-averaged precision and recall of the existing ARM technique are 47.9% and 60.9% respectively, whereas the macro-averaged for precision and recall of the SPE technique are 49.8% and 71.6% respectively. Thus the effectiveness of SPE is better than that of the ARM technique, recording improvements in macro-averaged precision and recall. However, our proposed model outperforms both benchmark techniques in precision, achieving a macro-averaged precision of 84.1%. Specifically, macro-averaged precision obtained by the proposed model is 34.3% and 36.2% higher than those reached by the existing ARM technique and SPE, respectively. The proposed model reaches to a macro-averaged recall at 66.2%, where improves the ARM by 5.3%, but it is about 5.4% less than SPE approach. When considering the micro average measures, we observe similar results to those we obtained by using macro average measures.

It is notable that we observe a more substantial improvement in precision that in recall with our proposed model and techniques. Observing from Table 4, our model makes significant improvements over others in all the datasets in precision, but in recall SPE has better performance. For example, our model records 36.2% and 34.3% improvements in terms of macro-averaged precision over the ARM and SPE techniques respectively, and 37.5% and 35% improvements in terms of micro-averaged precision. However, the proposed model achieves an averagely higher recall than the ARM technique but a slightly lower recall than the SPE technique. One reason is that for the iterative bootstrapping algorithm we limit number of output aspects between 70 and 120 aspects, therefore the precision for the output will be better than the recall, another reason for low recall is that of our model only works in detecting explicit aspects from review sentences.

Figure 3 shows the F-score measures of different approaches using different product datasets. In all five datasets, our model achieves the highest F-score. This

indicates our unsupervised model is effective in extracting correct aspects. We can thus draw the conclusion that our model is superior to the existing techniques, and can be used in practical settings, in particular those where high precision is required.
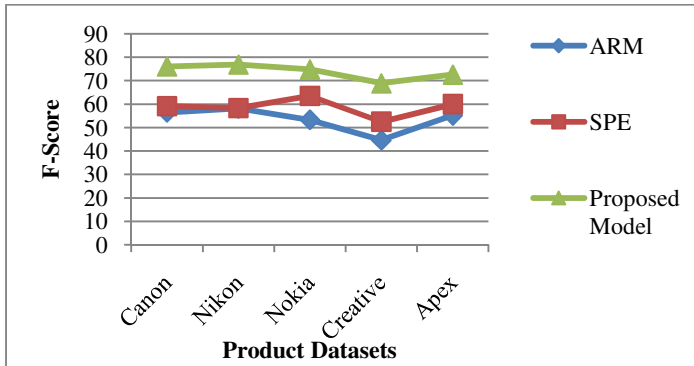

Fig. 3. F-scores of ARM, SPE, and the Proposed model for each dataset

This comparative evaluation suggests that the proposed model, which involves frequency-based and inter-connection information between the aspects and gives more importance to multi-word aspects and uses the influence of an opinion word in the review sentence, attains better effectiveness for product aspect extraction. The existing ARM technique depends on the frequencies of nouns or noun phrases for the aspect extraction, and SPE relies primarily on frequency- and semantic-based extraction of noun phrases for the aspect detection. For Example, our model is effective in detecting aspects such as "digital camera" or "battery charging system", which both ARM and SPE are failed on extraction of these non-noun phrases. Additionally, we can tune the parameters in our model to extracts aspects with less or more words, for example aspect "canon power shot g3" can be finding by the model. Finally, the results show using a completely unsupervised approach for aspect detection in sentiment analysis could achieve promising performances.

As mentioned before, the proposed model is an unsupervised domain-independent model. We therefore empirically investigate the performance of using a supervised technique for aspect detection in comparison to the proposed model. We employ results of a supervised technique from Somprasertsri and Lalitrojwong's work [8]. They proposed an approach for aspect detection by combining lexical and syntactic features with a maximum entropy model. Their approach uses the same data set collection of product reviews we experimented on. They extract the learning features from the annotated corpus of Canon G3 and Creative Labs Nomad Jukebox Zen Xtra 40 GB from customer review dataset. In their work, the set of data was split into a training set of 80% and a testing set of 20%. They employed the Maxent version 2.4.0 as the classification tool. Table 5 shows the micro-averaged precision, micro-averaged recall and micro-averaged F-score of their system output in comparison to our proposed model for the Canon and Creative datasets.

Table 4. Micro-averaged precision, recall and F-score for supervised maximum entropy and our unsupervised model

	Precision	Recall	F-score
Maximum entropy model	71.6	69.1	70.3
Proposed model	85.5	63.5	72.9

Table 5 shows that for the proposed model, the precision is improved dramatically by 13.9%, the recall is decreased by 5.6% and the F-score is increased by 2.6%. Therefore our proposed model and presented algorithms outperforms the Somprasertsri and Lalitrojwong's model. The significant difference between our model and theirs is that they use a fully supervised structure for aspect detection, but our proposed model is completely unsupervised and domain independent. Although in most applications the supervised techniques can achieve reasonable effectiveness, but preparing training dataset is time consuming and the effectiveness of the supervised techniques greatly depends on the representativeness of the training data. In contrast, unsupervised models automatically extract product aspects from customer reviews without involving training data. Moreover, the unsupervised models seem to be more flexible than the supervised ones for environments in which various and frequently expanding products get discussed in customer reviews.

5   Conclusions

This paper proposed a model for the task of identifying aspects in reviews. This model is able to deal with two major bottlenecks, domain dependency and the need for labeled data. We proposed a number of techniques for mining aspects from reviews. We used the inter-relation information between words in a review and the influence of an opinion word on detecting an aspect. Our experimental results indicate that our model is quite effective in performing the task. In our future work, we plan to further improve and refine our model. We plan to employ clustering methods in conjunction with the model to extract implicit and explicit aspects together to summarize output based on the opinions that have been expressed on them.

References

1. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Computational Linguistics 37(1), 9–27 (2011)
2. Thet, T.T., Na, J.C., Khoo, C.S.G.: Aspect-Based Sentiment Analysis of Movie Reviews on Discussion Boards. Journal of Information Science 36(6), 823–848 (2010)

3. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: American Association for Artificial Intelligence (AAAI) Conference, pp. 755–760 (2004)
4. Wei, C.P., Chen, Y.M., Yang, C.S., Yang, C.C.: Understanding what concerns consumers: A semantic approach to product feature extraction from consumer reviews. Information Systems and E-Business Management 8(2), 149–167 (2010)
5. Brody, S., Elhadad, N.: An unsupervised aspect-sentiment model for online reviews. In: 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, California, pp. 804–812 (2010)
6. Popescu, A., Etzioni, O.: Extracting product features and opinions from reviews. In: Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, pp. 339–346 (2005)
7. Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In: 3rd IEEE International Conference on Data Mining (ICDM 2003), Melbourne, FL, pp. 427–434 (2003)
8. Somprasertsri, G., Lalitrojwong, P.: Automatic product feature extraction from online product reviews using maximum entropy with lexical and syntactic features. In: IEEE International Conference on Information Reuse and Integration, pp. 250–255 (2008)
9. Zhu, J., Wang, H., Zhu, M., Tsou, B.K.: Aspect-based opinion polling from customer reviews. IEEE Transactions on Affective Computing 2(1), 37–49 (2011)
10. Zhai, Z., Liu, B., Xu, H., Jia, P.: Constrained LDA for Grouping Product Features in Opinion Mining. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part I. LNCS, vol. 6634, pp. 448–459. Springer, Heidelberg (2011)
11. Su, Q., Xu, X., Guo, H., Guo, Z., Wu, X., Zhang, X., Su, Z.: Hidden sentiment association in chinese web opinion mining. In: 17th International Conference on World Wide Web, Beijing, China, pp. 959–968 (2008)
12. Moghaddam, S., Ester, M.: ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews. In: 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 665–674. ACM (2011)
13. Fu, X., Liu, G., Guo, Y., Wang, Z.: Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon. Knowledge-Based Systems 37, 186–195 (2013)
14. Lin, C., He, Y., Everson, R., Ruger, S.: Weakly supervised joint sentiment-topic detection from text. IEEE Transaction on Knowledge & Data Engineering 24(6), 1134–1145 (2012)
15. Marcus, M., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics 19(2), 313–330 (1993)
16. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: Proceedings of HLT-NAACL, pp. 252–259 (2003)
17. Nakagawa, H., Mori, T.: Automatic Term Recognition based on Statistics of Compound Nouns and their Components. Terminology 9(2), 201–219 (2003)
18. Yoshida, M., Nakagawa, H.: Automatic Term Extraction Based on Perplexity of Compound Words. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 269–279. Springer, Heidelberg (2005)
19. Yang, Y.: An evaluation of statistical approaches to text categorization. Inf. Retr. 1(1-2), 69–90 (1999)

Cross-Lingual Natural Language Querying over the Web of Data

Nitish Aggarwal[1], Tamara Polajnar[2], and Paul Buitelaar[1]

[1] Unit for Natural Language Processing, Digital Enterprise Research Institute,
National University of Ireland, Galway
[2] Computer Laboratory, University of Cambridge, Cambridge
`firstname.lastname@deri.org`

Abstract. The rapid growth of the Semantic Web offers a wealth of semantic knowledge in the form of Linked Data and ontologies, which can be considered as large knowledge graphs of marked up Web data. However, much of this knowledge is only available in English, affecting effective information access in the multilingual Web. A particular challenge arises from the vocabulary gap resulting from the difference in the query and the data languages. In this paper, we present an approach to perform cross-lingual natural language queries on Linked Data. Our method includes three components: entity identification, linguistic analysis, and semantic relatedness. We use Cross-Lingual Explicit Semantic Analysis to overcome the language gap between the queries and data. The experimental results are evaluated against 50 German natural language queries. We show that an approach using a cross-lingual similarity and relatedness measure outperforms other systems that use automatic translation. We also discuss the queries that can be handled by our approach.

Keywords: Semantic Web, Natural Langauge Querying, CLIR.

1 Introduction

1.1 Motivation

In the last decade, the Semantic Web community has been working extensively towards creating standards, which tend to increase the accessibility of available information on the Web, by providing structured metadata[1]. Yahoo research recently reported [1] that 30% of all HTML pages on the Web contain structured metadata such as microdata, RDFa, or microformat. This structured metadata facilitates the possibility of automatic reasoning and inferencing. Thus, by embedding such knowledge within web documents, additional key information about the semantic relations among data objects mentioned in the web pages can be captured.

One of the most difficult challenge in multilingual web research is cross-lingual document retrieval, i.e. retrieval of relevant documents that are written in a language other than the query language. To address this issue we present a method for cross-lingual

---

[1] `http://events.linkeddata.org/ldow2012/slides/Bizer-LDOW2012-Panel-Background-Statistics.pdf`

natural language querying, which aims to retrieve all relevant information even if it is only available in a language different from the query language. Our approach differs from the state-of-the-art methods, which mainly consist of translating the queries into document languages ([2], [3]). However, the poor accuracy of automatic translation of short texts like queries makes this approach problematic. Hence, using large knowledge bases as an interlingua [4] may prove beneficial. The approach discussed here considers Linked Data as a structured knowledge graph. The Linked Open Data (LOD) cloud currently contains more than 291 different structured knowledge repositories in RDF[2] format, which are linked together using "DBpedia", "freebase" or "YAGO". It contains a large number of instances in many different languages, however, the vocabulary used to define ontology relations is mainly in English. Thus, querying this knowledge base is not possible in other languages even if the instances are multilingual. Cross-lingual natural language querying is required to access this structured knowledge base, which is the main objective of our approach.

1.2 Problem

Retrieval of structured data, in general, requires structured queries; however, effective construction of such queries is a laborious process. In order to provide a flexible querying environment, we propose to automatically construct a structured query from a natural language query (NL-Query). While there are several efforts ([5], [6], [7]) to convert a NL-Queries into structured SPARQL[3] queries in the monolingual scenario, the multilingual scenario offers further challenges. For example, the problem of mapping the query vocabulary to the ontology vocabulary is exacerbated by poor quality of automatic translation for short text and by the lack of multilingual structured resources. Therefore, to avoid relying on automatic translation, we present a novel approach for cross-lingual NL-Query formulation, which includes entity search, linguistic analysis, and semantic similarity and relatedness measure. We used Cross-Lingual Explicit Semantic Analysis (CL-ESA) to calculate the semantic relatedness scores between vocabularies in different languages.

1.3 Contribution

The main focus of our approach is the interpretation of NL-Queries by traversal over the structured knowledge graph, and the construction of a corresponding SPARQL query. As discussed in Section 1.2, translation based approaches for cross-lingual NL-Queries suffer from the poor quality of automatic translation. Therefore, in this paper, we introduce a novel approach for performing cross-lingual NL-Queries over structured knowledge base, without automatic translation. As an additional contribution, we have created and analyzed a benchmark dataset of 50 NL-Queries in German. We discuss the results of a comparison of our method with an automatic translation method over the 28 NL-Queries that can be addressed by our approach.

---

[2] Resource Description Framework (RDF) is the World Wide Web consortium (W3C) specification to represents the conceptual description. It was designed as a metadata data model.

[3] http://www.w3.org/TR/rdf-sparql-query/

Our algorithm can also be used for cross-lingual document retrieval provided that the document collection is already marked up with a knowledge base, for instance, Wikipedia articles annotated with DBpedia.

2   State of the Art

Most of the proposed approaches that address the task of Cross-Lingual Information Retrieval (CLIR) reduce the problem into a monolingual scenario by translating the search query or documents in the corresponding language. Many of them perform query translation ([8], [9], [2], [3])) into the language of the documents. However, all of these approaches suffer from the poor performance of machine translation on short texts (query). Jones et al. [3] performed query translation by restricting the translation to the cultural heritage domain, while Nquyen et al. [2] makes use of the Wikipedia cross-lingual links structure.

Without relying on machine translation, some approaches ([10], [11], [12]) make use of distributional semantics. They calculate a cross-lingual semantic relatedness score between the query and the documents. However, none of these approaches take any linguistic information into account, and do not make use of large available structured knowledge bases. With the assumption that documents of different languages are already marked-up with the knowledge base (for instance, Wikipedia articles are annotated with DBpedia), the problem of CLIR can be converted into querying over structured data. There is still a language barrier, as queries can be in different languages, while most of the structured data is only available in English. Qall-Me [13] performs NL-Querying over structured information by using textual entailment to convert a natural language question into SPARQL. This system relies on availability of multilingual structured data. It can only retrieve the information that is available in the query language. Therefore, this system is not able to perform CLIR. Freitas et al. [5] proposed an approach for natural language querying over linked data, based on the combination of entity search, a Wikipedia-based semantic relatedness (using ESA) measure, and spreading activation. Their approach is similar to ours, but it can not deal with different languages.

3   Background

3.1   DBpedia and SPARQL

We used DBpedia[4] as a knowledge base for our experiments. DBpedia is a large structured knowledge base, which is extracted from Wikipedia info-boxes. It contains data in the form of a large RDF graph, where each node represents an entity or a literal and the edges represent relations between entities. Each RDF statement can be divided into a subject, predicate and object. DBpedia contains a large ontology, describing more than 3.5 millions instances, forming a large general structured knowledge source. Also, it is very well-connected to several other Linked Data repositories in the Semantic Web. As

---

[4] `http://dbpedia.org/`

DBpedia instances are extracted from Wikipedia, they contains cross-links across the different languages, however, the properties (or relations) associated with the instances, are mainly defined in English.

In order to query DBpedia, a structured query is required. SPARQL is the standard structured query language for RDF, and allows users to write unambiguous queries to retrieve RDF triples.

3.2   Cross-Lingual ESA

Semantic relatedness of two given terms can be obtained by calculating the similarity between two high dimensional vectors in a distributed semantic space model (DSM). According to the distributional hypothesis, the semantic meaning of a word can (at least to a certain extent) be inferred from its usage in context, that is its distribution in text. This semantic representation is built through a statistical analysis over the large contextual information in which a term occurs. One recent popular model to calculate this relatedness by using the distributed semantics is Explicit Semantic Analysis (ESA) proposed by Gabrilovich and Markovitch [14], which attempts to represent the semantics of a given term in a high dimensional vector of explicitly defined concepts. In the original paper the Wikipedia articles were used to built the ESA model. Every dimension of the high dimensional vector reflects a unique Wikipedia concept or title, and the weight of the dimensions are created by taking the TF-IDF weight of a given term in the corresponding title of a Wikipedia document.

An interesting characteristic of Wikipedia is that this very large collective knowledge is available in multiple languages, which facilitates an extension of existing ESA for multiple languages called Cross-Lingual Explicit Semantic Analysis (CL-ESA) proposed by Sorg et al. [15]. The articles in Wikipedia are linked together across the languages. This cross-lingual linked structure can provide a mapping of a vector in one language to another. To understand CLESA, let us take two terms t_s in language L_s and t_t in language L_t. As a first step, a concept vector for t_s is created using the Wikipedia corpus in L_s. Similarly, the concept vector for t_t is created in L_t. Then, one of the concept vectors can be converted to the other language by using the cross-lingual links between articles across the languages, provided by Wikipedia. After obtaining both of the concept vectors in one language, the relatedness of the terms t_s and t_t can be calculated by using the cosine product, similar to ESA. For better efficiency, we chose to make a multilingual index by composing poly-lingual Wikipedia articles using the cross-lingual mappings. In such a case, no conversion of the concept vector in one language to the other is required. Instead, it is possible to represent the Wikipedia concept with some unique name common to all languages such as, for instance, the Uniform Resource Identifier (URI) of the English Wikipedia.

4   Approach

The key to our approach is the interpretation of NL-Queries in different languages, by using a combination of *entity identification, linguistic analysis* and *cross-lingual similarity and relatedness measure*. Figure 1 shows the three components of our approach

along with an example of a NL-Query in German[5]. The interpretation process starts with the identification of possible entities appearing in a given NL-Query, followed by linguistic analysis of the NL-Query. The system performs the whole pipeline with all of the identified entities and takes union over all of the retrieved results. Using the dependencies provided by the linguistic analysis, our system determines the next term that will be compared with all the relations associated with the identified entity, to find the best matched relation. For instance, in example shown in Figure 1, the system identified "Bill Clinton" as entity and "Tochter" as next term. Following the process, it calculates the similarity score with every relation associated with "Bill Clinton" and finds the maximum similar relation to obtain the next entity from the knowledge base.

4.1   Entity Identification

The first step of the interpretation process is the identification of potential entities, i.e. the Linked Data concepts (classes and instances), present in the NL-Query. A baseline entity identification can be defined as the identification of an exact match between the label of a concept against the term appearing in the NL-Query; for example, DBpedia: Bill_Clinton shown in Figure 1. "Bill Clinton" is the name of a person and it appeared as a label of DBpedia: Bill_Clinton URI in the database. However, a term such as "Ministerpräsidenten von Spanien" and "Christus im Sturm auf dem See Genezareth" do not appear as labels in the database. Therefore, in order to resolve this issue, we translate the term to get the approximate term in the corresponding language and find the best matched label in the database. We use the Bing translation system[6] to perform the automatic translation but the quality of translation is not very promising and we do not get the exact translation of a given label. Therefore, we calculate the token edit distance between translated label and the labels in our database and select the maximum matched one. For instance, the translation of "Christ in the storm on the sea of Galilee" is "Christ in storm on the sea of Galilee" but label of the appropriate concept is "The Storm on the Sea of Galilee".

In addition, our approach includes a disambiguation process, in which we disambiguate the selected concept candidates based on their associated relations in the knowledge base. For instance, in a given NL-Query "Wie viele Angestellte hat Google?"@de[7] two different DBpedia entities can be found with the label "Google", i.e. "DBpedia: Google_Search" and "DBpedia: Google". We calculate similarity scores with all associated relations of both, and find that term "Angestellte" in the NL-Query obtained maximum similarity score with the relation "number Employees", which is associated with "DBpedia: Google".

---

[5] Translated from the QALD-2 challenge dataset, which has 100 NL-Queries in English, over DBpedia.

[6] http://www.bing.com/translator

[7] Translation of "How many employees does Google have?" from the English test dataset.
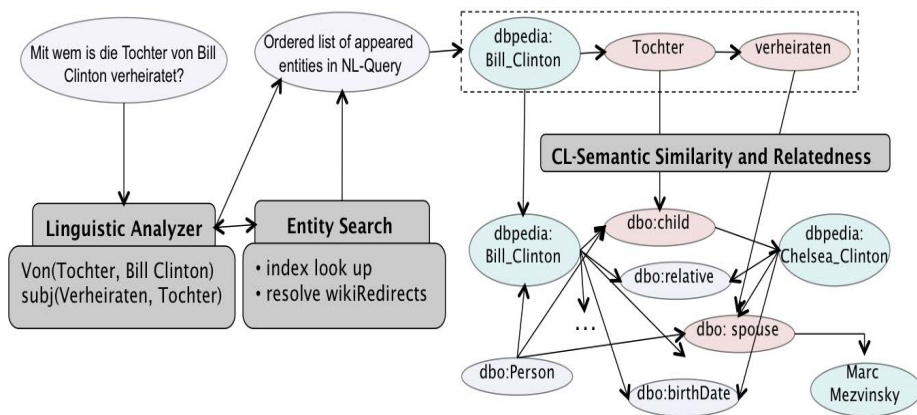
Fig. 1. Query interpretation pipeline for the German NL-Query "Mit wem is die Tochter von Bill Clinton verheiratet?" ("Who is the daughter of Bill Clinton married to?"@en)

4.2 Linguistic Analysis

Linguistic analysis of the NL-Queries is needed to get the dependencies among the identified entities and terms. We use the Stanford parser[8] for German to generate the dependencies. Following these dependencies, we convert the given NL-Query into a Direct Acyclic Graph (DAG). Vertices of the generated DAG represent the entities and edges reflect the terms directly dependent on the obtained entities (vertices). Figure 1 shows the DAG obtained from our example query "Mit wem is die Tochter von Bil Clinton verheiratet?". To generate the DAG, first we obtain the central entity from the previous step. With the relations of this central entity, semantic matching will be performed. Therefore, we retrieve the directly dependent terms of the central entity by following the generated Stanford typed dependencies, and add them into the DAG. Similarly, we perform this action for all the other terms in the list. For instance, in our example NL-Query shown in Figure 1, firstly, the system identifies "Bill Clinton" as a central entity,[9] and then "Tochter" as direct dependent of "Bill Clinton" followed by "verheiratet" as direct dependent of "Tochter".

4.3 Knowledge Graph Traversing Using Semantic Similarity and Relatedness

A knowledge graph can be defined as the structured data of well-connected entities and their relations. Our next step is to find such relations of selected central entity in the knowledge base that are best matches with the term directly dependent on this central entity in the generated DAG. First, we search for the entity "Bill Clinton" in DBpedia as our approach takes DBpedia as knowledge base, and retrieve all of the relations (DBpedia properties) associated with it. Then, we find the best semantically match DBpedia

---

[8] http://nlp.stanford.edu/software/lex-parser.shtml
[9] The term to start the search around in the whole DBpedia graph.

property of the direct dependent term "Tochter" by calculating a cross-lingual similarity score between all the DBpedia properties of Bill Clinton and "Tochter". After obtaining the relevant property, i.e. "child", we find the entity DBpedia: Chelsea_Clinton, connected with entity "DBpedia: Bill_Clinton" by property "child". We perform the same steps with the retrieved entity for the directly dependent term "verheiratet" of "Tochter", and so on until the end of the DAG. Finally, we retrieve the most relevant entity and all the associated documents in different languages containing a description about this entity.

5　Evaluation

5.1　Datasets

In order to evaluate our approach, we created a testset of 50 NL-Queries in German. The benchmark is created by manually translating the English NL-Queries provided by the "Question Answering over Linked Data (QALD-2)" dataset, consists of 100 NL-Queries in English over DBpedia. All of the NL-Queries are annotated with keywords, corresponding SPARQL queries and answers retrieved from DBpedia. Also, every NL-Query specifies some additional attributes, for example, if a mathematical operation such as aggregation, count or sort is needed in order to retrieve the appropriate answers.

Table 1. Query categorization of training and test dataset

Dataset	Simple	Template-based	SPARQL aggregation
Training	27	11	12
Test	28	10	12

We translated QALD-2 dataset and divided it into two parts, one for training and one for testing. Therefore, each dataset contains 50 NL-Queries in German. We performed a manual analysis to keep the same complexity level in both the datasets. We divided all of the NL-Queries into three different categories: simple, template-based and SPARQL aggregation. Simple queries contain the DBpedia entities and their relations (DBpedia properties), and do not need a predefined template or rule to construct the corresponding SPARQL query. However, these queries include semantic and linguistic variations, that means they express the DBpedia properties by using related terms rather than having the exact label of a property. For instance, in a given query "How tall is Michael Jordan?", "tall" does not appear in the vocabulary of DBpedia properties, however, the answer of the query can be retrieved by DBpedia property "height" appearing with "DBpedia: Michael_Jordan". Those queries, which required predefined templates or rules, are categorized as template-based [6] queries, for example, the query "Give me all professional skateboarders from Sweden." required a predefined template for retrieving all persons with occupation Skateboarding and born in Sweden. SPARQL aggregation type of queries need performing a mathematical operation such as aggregation, count or sort, therefore, they also require a predefined template.

Following the categorization, we divided the dataset into two parts by keeping an equal number of queries in each category. We then performed our experiments on the prepared test dataset of 50 NL-Queries in German. Table 1 shows the statistics about both the datasets. We are extending these datasets for other languages and they are freely available.

Table 2. Error type and its distribution over 50 natural language queries and 28 selected natural language queries in German

Error Type	No of NL-Queries	
	out of 50	out of 28
Entity Identification without Translation	10	3
Entity Identification with Translation	7	1
Linguistic Analysis	14	4
At least one	18	5

5.2   Experiment

We evaluated the outcome of our approach at all three stages of the processing pipeline: 1) entity identification, 2) linguistic analysis, and 3) semantic similarity and relatedness measures. This way, we can investigate the errors introduced by individual components. As shown in Figure 1, the third component "semantic similarity and relatedness measures" relies on the correctness of the constructed DAG, i.e. on the performance of both the previous components (entity identification and linguistic analysis). Therefore, it is important to examine the performance of individual components. We evaluated the outcome of entity identification and linguistic analysis on all 50 NL-Queries of the test dataset. However, all of the template-based and SPARQL aggregation type NL-Queries are out of the scope of our settings. Therefore, we discuss the results obtained for remaining 28 NL-Queries. The entity identification component was evaluated in both ways; entity identification without using automatic translation and entity identification with automatic translation[10]. Table 2 shows that appropriate entities could not be found in 10 NL-Queries out of 50 NL-Queries and 3 NL-Queries out of 28. However, by using automatic translation the error is reduced to 7 and 1 NL-Queries respectively. To evaluate the performance of the linguistic analysis component, we counted the number of NL-Queries, for which the Stanford parser was unable to generate the dependencies. The statistics of the errors in linguistic analysis are shown in Table 2. As explained in Section 4.3, to find the relevant properties associated with the selected DBpedia entity, a comparison of all the properties and the next term from the DAG is needed. This requires a good cross-lingual similarity and relatedness measure. Therefore, to examine the effect of similarity and relatedness measure over automatic translation, we used three different settings in calculating the scores: a) automatic translation followed by

---

[10] The automatic translation was only used for those entities, that could not be found in the database with the given labels.

Table 3. Evaluation on 28 German NL-Queries

NL-Queries in German and English	Translation with edit dist.			Tranlation with ESA			CL-ESA		
	P	R	F1	P	R	F1	P	R	F1
1. Wer war der Nachfolger von John F. Kennedy?@de Who was the successor of John F. Kennedy?@en	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2. Wie viele Studenten hat die Freie Universität Amsterdam?@de How many students does the Free University in Amsterdam have?@en	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3. Über welche Länder erstreckt sich das Himalaya-Gebirgssystem?@de To which countries does the Himalayan mountain system extend?@en	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4. Gib mir alle Mitglieder von The Prodigy.@de Give me all members of Prodigy.@en	0.0	0.0	0.0	1.0	0.28	0.44	1.0	0.28	0.44
5. Wie groß ist Michael Jordan?@de How tall is Michael Jordan?@en	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0
6. Wer ist der Gouverneur von Texas?@de Who is the governor of Texas?@en	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7. Sean Parnell ist der Gouverneur welches US-Bundesstaates?@de Sean Parnell is the governor of which U.S. state?@en	0.33	1.0	0.5	0.33	1.0	0.5	0.33	1.0	0.5
8. Welches ist der Geburtsname von Angela Merkel?@de What is the birth name of Angela Merkel?@en	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
9. Wie oft hat Nicole Kidman geheiratet?@de How often did Nicole Kidman marry?@en	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0
10. Wer hat Skype entwickelt?@de Who developed Skype?@en	1.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0
11. Gib mir alle Partnerstädte von Brünn.@de Give me all sister cities of Brno.@en	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0
12. Wer hat Intel gegründet?@de Who founded Intel?@en	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
13. Gib mir alle Rassen des Deutscher Schäferhund.@de Give me all breeds of the German Shepherd dog.@en	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0
14. Wer hat den Reissverschluss erfunden?@de Who invented the zipper?@en	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0
15. Welche Länder sind durch den Rhein verbunden?@de Which countries are connected by the Rhine?@en	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
16. In welcher britischen Stadt ist der Hauptsitz des MI6?@de In which UK city are the headquarters of the MI6?@en	0.5	1.0	0.66	0.5	1.0	0.66	0.5	1.0	0.66
17. Welches sind die Spitznamen von San Francisco?@de What are the nicknames of San Francisco?@en	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0
18. Gib mir die Astronauten von Apollo 14.@de Give me the Apollo 14 astronauts.@en	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19. Wie viele Kinder hatte Benjamin Franklin?@de Which ships were called after Benjamin Franklin?@en	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
20. Welche Instrumente hat John Lennon gespielt?@de Which instruments did John Lennon play?@en	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0
21. Wie viele Angestellte hat Google?@de How many employees does Google have?@en	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
22. Wann ist Michael Jackson gestorben?@de When did Michael Jackson die?@en	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23. Wie hoch ist der Mount Everest?@de How high is the Mount Everest?@en	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24. Mit wem is die Tochter von Bill Clinton verheiratet?@de Who is the daughter of Bill Clinton married to?@en	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0
24. Wer hat die Musik für Harold und Maude komponiert?@de Who composed the music for Harold and Maude?@en	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0
26. Wo ist die Residenz des Ministerpräsidenten von Spanien?@de Where is the residence of the prime minister of Spain?@en	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
27. Aus welchem Land kommt der Schöpfer von Nijntje?@de Which country does the creator of Miffy come from?@en	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0
28. Wer malte Christus im Sturm auf dem See Genezareth?@de Who painted The Storm on the Sea of Galilee?@en	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0
Combined	**0.387**	**0.429**	**0.407**	**0.601**	**0.617**	**0.609**	**0.815**	**0.831**	**0.823**

string edit distance, b) automatic translation followed by monolingual Explicit Semantic Analysis (ESA), and c) Cross-Lingual Explicit Semantic Analysis (CL-ESA). To evaluate the performance of automatic translation over CL-ESA, we reduce the problem into a monolingual scenario, by translating the properties into corresponding language, in settings a and b. Automatic translation is not performed on the full text of a NL-Query but only on the properties because the quality of translation is not good enough to get the correct linguistic dependencies by using Stanford parser.

In first setting, we perform the translation and check if we can find the translated term in the listed properties by using Levenshtein edit distance approximation. While, in the second one, we calculate similarity and relatedness scores using ESA after performing automatic translation, to investigate if the automatic translation and semantic relatedness can complement each other. We do not use automatic translation in the third setting "cross-lingual semantic similarity and relatedness measure", but only rely on the scores generated by CL-ESA. The quality of final results generated by all three settings are analyzed manually and shown in Table 3 and we discuss it in detail in next Section 5.3.

5.3   Results and Discussion

Table 3 compares the results obtained by using three different settings of our approach: a) automatic translation followed by Levenshtein edit distance, b) automatic translation followed by monolingual ESA, and c) CL-ESA. It shows that automatic translation can not bridge the vocabulary gap between NL-Queries and DBpedia. That means there are large lexical variations in defining the relations of entities. Further, we investigate if automatic translation and monolingual ESA can complement each other. Although, the overall score generated by the combination of both is improved significantly over the score obtained by using automatic translation, the best results are generated by CL-ESA. The reason may be that combined errors introduced by using both translation and ESA is more than the error generated by CL-ESA.

Table 2 shows that 5 NL-Queries out of these 28 NL-Queries pose at least one type of error (entity identification or linguistic analysis), meaning that DAGs can be generated only for 23 NL-Queries out of 28. However, to reduce this error, we consider that keywords appearing in a given NL-Query may depend on the selected entity. For instance, the Stanford parser failed to generate the correct dependencies for Q28 and Q5 (listed in Table 3) but by considering the terms "groß" and "malte" to be dependent on the identified entities "Michael Jordan" and "Christus im Sturm auf dem See Genezareth" respectively, we could generate the correct DAGs. Therefore, we can test the third component of our approach on 25 NL-Queries out of 28 as we got the correct DAGs of 25 NL-Queries. We can see in Table 3, in our approach, by using translation we can retrieve the correct answers for 10 NL-Queries and partially correct for 2 NL-Queries; by using translation with the combination of ESA we can retrieve the correct answers for 15 NL-Queries and partially correct for 3 NL-Queries; by using CL-ESA we can retrieve the correct answers for 21 NL-Queries and partially correct for 3 NL-Queries.

In case of Q10, automatic translation followed by ESA failed because when the system tried to find the maximum related property with the term "developed" (translation of "entwickelt"), it obtained a higher ESA score for another property "operating

system" than "developer". Our approach simply failed to find the results for Q14, due to the appearance of more than one highly related properties, such as "mission name", "mission duration", "mission" and "launch pad", for identified entity "Apollo 14", with "Astronauten" and "astronauts".

Our approach can also retrieve the partial set of appropriate results for more complex NL-Queries like "Gib mir alle Menschen, die in Wien geboren wurden und in Berlin gestorben sind"[11]. Therefore, we also report the performance of our system on the overall test dataset of 50 NL-Queries. The results are shown in Table 4. In this way, we can find the overall coverage of our approach on all types of NL-Queries.

Table 4. Evaluation on 50 German NL-Queries

	Average Precision	Average Recall	F1
Translation	0.217	0.24	0.228
Translation with ESA	0.34	0.386	0.361
CL-ESA	0.459	0.506	0.481

6   Conclusion and Future Work

This paper presented a novel approach to perform cross-lingual natural language querying over Linked Data that includes *entity search, linguistic analysis* and *cross-lingual semantic similarity and relatedness measure*. The approach was evaluated against 50 NL-Queries in German over DBpedia and achieved an average precision of 0.459, an average recall of 0.506 and F1 score of 0.361. However, on the NL-Queries that can be covered by this approach, the system achieved an average precision of 0.815, an average recall of 0.831 and a F1 score of 0.823. Our approach clearly shows that cross-lingual semantic similarity and relatedness measures outperform the automatic translation for cross-lingual NL-Querying over Linked Data.With this approach, cross-lingual information retrieval at document level can also be performed, if the documents are already marked up with the structured knowledge base. Therefore, we are planning to extend our approach for entity retrieval and associated documents, and evaluate the approach in the traditional information retrieval scenario.

References

1. Mika, P., Potter, T.: Metadata statistics for a large web corpus. In: WWW 2012 Workshop on Linked Data on the Web (2012)
2. Nguyen, D., Overwijk, A., Hauff, C., Trieschnigg, D.R.B., Hiemstra, D., De Jong, F.: Wikitranslate: query translation for cross-lingual information retrieval using only wikipedia. In: Proceedings of the 9th CLEF (2009)

---

[11] "Give me all people that were born in Vienna and died in Berlin." in the English test dataset.

3. Jones, G., Fantino, F., Newman, E., Zhang, Y.: Domain-specific query translation for multilingual information access using machine translation augmented with dictionaries mined from Wikipedia. In: CLIA 2008, p. 34 (2008)

4. Steinberger, R., Pouliquen, B., Ignat, C.: Exploiting multilingual nomenclatures and language-independent text features as an interlingua for cross-lingual text analysis applications. In: Proc. of the 4th Slovenian Language Technology Conf., Information Society (2004)

5. Freitas, A., Oliveira, J.G., O'Riain, S., Curry, E., Pereira da Silva, J.C.: Querying linked data using semantic relatedness: a vocabulary independent approach. In: Muñoz, R., Montoyo, A., Métais, E. (eds.) NLDB 2011. LNCS, vol. 6716, pp. 40–51. Springer, Heidelberg (2011)

6. Unger, C., Bhmann, L., Lehmann, J., Ngomo, A.C.N., Gerber, D., Cimiano, P.: Sparql template based question answering. In: 21st International World Wide Web Conference, WWW 2012 (2012)

7. Yahya, M., Berberich, K., Elbassuoni, S., Ramanath, M., Tresp, V., Weikum, G.: Natural language questions for the web of data. In: EMNLP-CoNLL 2012 (2012)

8. Lu, C., Xu, Y., Geva, S.: Web-based query translation for english-chinese CLIR. In: Computational Linguistics and Chinese Language Processing (CLCLP), pp. 61–90 (2008)

9. Pirkola, A., Hedlund, T., Keskustalo, H., Järvelin, K.: Dictionary-based cross-language information retrieval: Problems, methods, and research findings. Information Retrieval, 209–230 (2001)

10. Littman, M., Dumais, S.T., Landauer, T.K.: Automatic cross-language information retrieval using latent semantic indexing. In: Cross-Language Information Retrieval, ch. 5, pp. 51–62. Kluwer Academic Publishers (1998)

11. Zhang, D., Mei, Q., Zhai, C.: Cross-lingual latent topic extraction. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL 2010, pp. 1128–1137. Association for Computational Linguistics, Stroudsburg (2010)

12. Sorg, P., Braun, M., Nicolay, D., Cimiano, P.: Cross-lingual information retrieval based on multiple indexes. In: Working Notes for the CLEF 2009 Workshop, Cross-Lingual Evaluation Forum, Corfu, Greece (2009)

13. Ferrández, Ó., Spurk, C., Kouylekov, M., Dornescu, I., Ferrández, S., Negri, M., Izquierdo, R., Tomás, D., Orasan, C., Neumann, G., Magnini, B., Vicedo, J.L.: The qall-me framework: A specifiable-domain multilingual question answering architecture. Web Semantics, 137–145 (2011)

14. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 1606–1611 (2007)

15. Sorg, P., Cimiano, P.: An experimental comparison of explicit semantic analysis implementations for cross-language retrieval. In: Horacek, H., Métais, E., Muñoz, R., Wolska, M. (eds.) NLDB 2009. LNCS, vol. 5723, pp. 36–48. Springer, Heidelberg (2010)

Extractive Text Summarization:
Can We Use the Same Techniques for Any Text?

Tatiana Vodolazova, Elena Lloret, Rafael Muñoz, and Manuel Palomar

Dept. Lenguajes y Sistemas Informáticos, Universidad de Alicante
Apdo de correos, 99, E-03080, Alicante, Spain
{tvodolazova,elloret,rafael,mpalomar}@dlsi.ua.es

Abstract. In this paper we address two issues. The first one analyzes whether the performance of a text summarization method depends on the topic of a document. The second one is concerned with how certain linguistic properties of a text may affect the performance of a number of automatic text summarization methods. For this we consider semantic analysis methods, such as textual entailment and anaphora resolution, and we study how they are related to proper noun, pronoun and noun ratios calculated over original documents that are grouped into related topics. Given the obtained results, we can conclude that although our first hypothesis is not supported, since it has been found no evident relationship between the topic of a document and the performance of the methods employed, adapting summarization systems to the linguistic properties of input documents benefits the process of summarization.

Keywords: text summarization, textual entailment, anaphora resolution.

1 Introduction

The first attempts to tackle the task of automatic text summarization were made as early as in the middle of the past century [17]. Since then the capabilities of modern hardware have increased enormously. However, nowadays when we talk about automatic text summarization we mostly focus on extractive summaries and hope to top the threshold of 50% on the recall [22]. Extractive summaries, as opposed to the abstractive ones that involve natural language generation techniques, consist of segments of the original text. The task sounds less challenging than it has been proven to be [22].

The extractive summarization systems developed so far have been tested on a number of different corpora [22]. There has been a significant number of systems proposed for the task of summarization of the newswire articles. Many of those systems emerged due to the Document Understanding Conference challenges (DUC)[1] [8,25]. Even that the last challenge has been held in 2007, the DUC data is still being used in research [15,16,24,26]. Some experiments were done

---

[1] http://duc.nist.gov/

with the Reuters newswire corpus [2]. The short newswire articles differ from fiction. The summarization systems that target this niche have adapted to the particular characteristics of fiction. There has been research on short fiction summarization [12], fairy tales [16], whole books [19], etc. Due to the rapid growth of the amounts of web data, the need to summarize becomes even more acute. More recent research has focused on Web 2.0 textual genres, such as forum [30] and blog [11] summarization. The specific language used in blogs and forums makes the task being different to that of newswire article summarization. Between the blog and the newswire summarization we could place the e-mail summarization that ranges from summarizing a single e-mail message [20] to the whole thread of related e-mails [23]. Automatic text summarization has also been combined with speech recognition to summarize spoken dialogues [9,18].

Summarization systems have been adapted to a number of different domains. In particular, there has been an extensive research in summarizing medical documents [1]; a) medical journal articles [6,3]; b) healthcare documents for patients [7]. Another domain that attracted the attention is the legal domain. There have been some experiments with the documents from the European Legislation Website[2] [3].

However, text documents differ depending on genre, text type, domain, sublanguage, style, particular topic covered, etc. (for a detailed discussion see [13]). Personal style of a writer, their vocabulary size, word choice, use of expressive means and irony, sentence length and structure preferences are not less affecting. Dialogues and monologues, science fiction and love stories, technical reports and newswire articles, poems and legalese, use of metaphors and synonyms, anaphoric expressions and proper nouns all these carry with them their unique properties. Those properties may affect the quality of summaries generated using the techniques developed for the task of automatic text summarization. And in this paper we would like to study this issue.

We adapt out systems to specific domains, genres, text styles. We develop and implement different summarization techniques and heuristics. But to the best of our knowledge, so far there has been no attempt to treat documents in a collection differently from each other. If a system makes use of pronominal anaphora resolution module, it will try to resolve anaphora in all the documents. Now, what if the document contains only a few pronouns? The performance will slow down but the results will stay the same. What if a document contains a high number of pronouns and the chosen anaphora resolution module cannot handle them correctly? The performance will slow down and the resulting summary will be of a worse quality. If we consider word sense disambiguation task and some specific domains like e.g. legalese documents, the language used is so precise that synonymy disambiguation will probably introduce no improvement into the quality of summaries.

In this paper we address two issues. The first one is concerned with the problem of preliminary document analysis and how the linguistic properties of a text may affect the performance of a number of automatic text summarization

---

[2] `http://eur-lex.europa.eu/en/legis/index.htm`

techniques. We have focused on the basic linguistic characteristics of text, such as the noun ratio, pronoun ratio and personal noun ratio. And we have analyzed how they affect the summarization systems that use textual entailment and anaphora resolution tools to aid in the summarization process. The final goal would be to develop a system that chooses the best summarization techniques based on the linguistic properties of a document. Moreover, we have divided our corpus in groups according to the topic covered. The second goal of this research is to analyze whether the performance of a text summarization engine depends on the topic of a document.

This paper is structured as follows. Section 2 reports on the related work. Section 3 describes in detail the system used for the experiment. The corpus is described in Section 4. The results are discussed in Section 5. Finally, conclusion and future work are outlined in Section 6.

2   Related Work

With the evolution of technology different methods and heuristics have been used to improve extractive summarization systems. The early systems relied on the simple heuristics: i) *sentence location* (sentences located in the beginning or end of the text, headings and the sentences highlighted in bold among others are considered to be more important and are included in the final summary) [5]; ii) *cue phrases* (presence of previously defined words and phrases as "concluding", "argue", "propose" or "this paper") [5,28]; iii) *segment length* (sentences with the length below some predefined threshold can be automatically ignored) [28]; iv) *the most frequent words* (exploring the term distribution of a document allows to identify the most frequent words that are assumed to represent at the same time the most important concepts of the document) [17].

Today we apply various methods to structure information that we extract from documents and to analyze it intelligently. Graph theory has been successfully applied to represent the semantic contents of a document [24] . Latent Semantic Analysis, that involves term by sentences matrix representation and singular value decomposition has also been proven to benefit the task of extractive summarization [26,10]. A number of machine learning algorithms such as decision trees, rule induction, decision forests, Naive Bayes classifiers and neural networks among others have been adapted to this task as well [20,4]. Part-of-speech taggers [20], word sense disambiguation algorithms [24], anaphora resolution [26], textual entailment [27,15] and chunking [20] are among the most frequently used linguistic analysis methods.

To the best of our knowledge there has been no attempt to analyze the impact of shallow linguistic properties of the original text on the quality of automatically generated summaries.

However, there has been a related work involving automatic text summarization and sentence structure. Nenkova et al. [21] focused on how sentence structure can help to predict the linguistic quality of generated summaries. The authors selected a set of structural features that include:

- *sentence length*
- *parse tree depth*
- *number of fragment tags in the sentence parse*
- *phrase type proportion*
- *average phrase length*
- *phrase type rate* was computed for prepositional, noun and verb phrases as diving the number of words of each phrase by the sentence length
- *phrase length* was computed for prepositional, noun and verb phrases as diving the number of phrases of the given type that appeared in the sentence by the sentence length
- *length of NPs/PPs contained in a VP*
- *head noun modifiers*

Though the set of features is different and more diverse, the phrase type ratio and phrase length can be probably compared to the noun, pronoun and proper noun ratios selected for our research. A ranking SVM was trained using these features. The summary ranking accuracy of the ranking SVM was compared to other linguistic quality measures, that include Coh-Metrix, language models, word coherence and entity coherence measures. The evaluation of results was done on the system and input levels. Whereas in the former all participating systems were ranked according to their performance on the entire test set, and in the latter all the summaries produced for a single given input.

Structural features proved to be best suitable for input-level human summaries, middle of the range for input level system summaries and about the worst class of features for system-level evaluation of automatic summaries. At the same time being the most stable set of features and ranging the least across the chosen evaluation settings.

3   Summarization System

To analyze the impact of proper noun, pronoun and noun ratios we have chosen the summarization system described in [29]. The system allows a modular combination of anaphora resolution, textual entailment and word sense disambiguation tools with the term or concept frequency based scoring module. In this research we focused on textual entailment and anaphora resolution.

Textual Entailment. The task of textual entailment is to capture the semantic inference between text fragments. There has been a number of summarization systems utilizing textual entailment to aid in summarization process. Both in the process of evaluating the final summary and in the process of summary generation. In the latter case textual entailment is often applied to eliminate the semantic redundancy of a document [15].

Anaphora resolution. Powerful pronominal anaphora resolution tool relates pronouns to their nominal antecedents. This is of use to all the summarization methods that rely on term overlap, from the simple term frequency to latent semantic analysis. Steinberger et al. [26] report an increase of 1.5% for their

summarization system based on latent semantic analysis when anaphoric information is included.

The system that we have chosen for this experiment consists of 4 modules: anaphora resolution, textual entailment, word sense disambiguation and scoring modules (see Figure 1). The scoring module is essential. It is the last step in the process when the final sentence scores are calculated. The remaining 3 modules can be freely combined pairwise with each other and/or with the scoring module. All the 4 modules can be applied at once as well. This suits well our purpose of analyzing the impact of different shallow linguistic properties of a text on various kinds of summarization techniques, that are represented by modules in our case. For this study we do not use all the possible combinations of modules (for the precise list of combinations see Section 5.1)
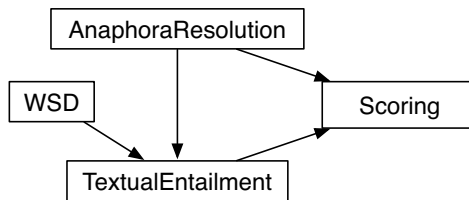

Fig. 1. Interaction of semantic components

4   Topics and Linguistic Properties of the Data Set

Out data is a set of newswire articles, taken from the Document Understanding Conference challenge of 2002. The original set consists of about 530 articles, that are grouped topicwise into a set of 59 subgroups. The original grouping involves some duplicate articles and there are some topics/events that are represented by more than one such group.

One of the goals of this research is to investigate whether the topic of a document affects the quality of a generated summary. Due to that fact we have selected about 270 articles. The duplicates were removed. The articles were manually reviewed and grouped trying to keep the original DUC grouping whenever possible. Below is the list of the resulted topics. The number of documents in each group is stated in the round brackets.

1. battleship explosion (11)
2. ferry accidents (9)
3. IRA attack (8)
4. earthquake Iran (15)
5. China flood (10)
6. Hurricane Gilbert (13)
7. Mount Pinatubo volcano (5)
8. North American drought (9)
9. thunderstorm US (11)
10. Checkpoint Charlie (5)
11. abortion law (6)
12. Germany reunification (14)
13. Honecker protest (11)
14. Iraq invades Kuwait (27)

15. Robert Maxwell companies (10)
16. striking coal miners (12)
17. US ambassadors (11)
18. Super Bowl (10)
19. marathon (9)
20. Olympics (10)

21. Leonard Bernstein (13)
22. Lucille Ball (14)
23. Margaret Thatcher (10)
24. Sam Walton (7)
25. Gorbachev (10)

We have further grouped the selected articles according to the more general topic covered, e.g. *marathon*, *Olympics*, *Super Bowl* were assigned to the topic on *sports*, etc. This yielded 5 groups, covering the general topics on *accidents*, *natural disasters*, *politics*, *sports* and *famous people* (please see Table 1 for more details).

Having grouped the data in different topics, we proceeded with their linguistic analysis. The selected documents were processed using a part-of-speech tagger to obtain the average *noun* (NR), *pronoun* (PR) and *proper noun ratios* (PNR) for each of the 25 topics. These ratios were calculated by diving the number of words of the respective word class by the total number of words in a document.

Table 1. Linguistic properties of the original documents

		PNR	NR	PR	size
accidents	1. battleship explosion	0.11466	0.34381	0.03540	**670.0**
	2. ferry accidents	0.10874	0.34006	0.04024	423.666
	3. IRA attack	0.10666	0.31853	**0.05897**	599.625
natural	4. earthquake Iran	**0.15052**	**0.35761**	0.02933	444.8
disasters	5. China flood	0.12880	**0.38535**	0.02032	383.8
	6. Hurricane Gilbert	**0.13867**	**0.36818**	0.02501	**730.923**
	7. Mount Pinatubo volcano	0.10716	0.33642	0.03041	**672.4**
	8. North American drought	0.10402	0.34050	0.02485	398.0
	9. thunderstorm US	0.13165	**0.36461**	0.02791	**718.7**
politics	10. Checkpoint Charlie	**0.16148**	0.34312	**0.04406**	513.2
	11. abortion law	0.11954	0.33815	**0.06449**	545.833
	12. Germany reunification	**0.14561**	0.33543	0.03163	558.5
	13. Honecker protest	**0.15005**	0.34585	0.03887	286.545
	14. Iraq invades Kuwait	**0.16509**	**0.36821**	0.03189	552.555
	15. Robert Maxwell companies	**0.17074**	**0.37558**	0.03782	444.1
	16. striking coal miners	0.09267	**0.36252**	0.02651	507.083
	17. US ambassadors	**0.20187**	**0.38561**	0.03811	415.545
sports	18. Super Bowl	**0.17758**	**0.39363**	0.03184	438.8
	19. marathon	0.13454	0.33495	**0.05224**	**810.555**
	20. Olympics	**0.15359**	**0.35780**	0.04059	**607.5**
famous	21. Leonard Bernstein	**0.19529**	**0.38679**	**0.04427**	596.923
people	22. Lucille Ball	0.13095	0.32332	**0.08895**	848.714
	23. Margaret Thatcher	0.13329	0.31561	**0.06571**	**624.4**
	24. Sam Walton	0.10693	0.32362	**0.05967**	566.714
	25. Gorbachev	0.09943	0.31251	**0.05673**	**745.9**
average		0.13718	0.35031	0.04183	564.191

This shallow linguistic analysis methods were chosen in agreement with the summarization system described in Section 3. The noun ratio has been chosen since a topic of a document is usually characterized in the form of noun phrases and textual entailment (with or without the word sense disambiguation) can be used to eliminate the semantic redundancy. The anaphora resolution process involves analyzing the pairs of nouns, pronouns and proper nouns in a document.

Table 1 contains the results for the selected features topic-wise. The figures higher than the average are highlighted in bold. Already on this shallow analysis level it can be seen, that different topics have different tendencies. The documents that cover political issues and sports tend to have a higher number of proper nouns. The articles about famous people contain a lot of pronouns. The latter led us to the hypothesis, that summarization systems that involve anaphora resolution would yield summaries of a better quality for those articles. While the former suggested to rather apply a textual entailment heuristics. The actual results obtained when applying the selected summarization system to the set of 25 groups of documents are discussed in Section 5.

5   Results and Discussion

5.1   Experiment Setup

The chosen summarization system as explained in Section 3 (see Figure 1) allows to freely combine anaphora resolution, textual entailment, word sense disambiguation and scoring modules. This suits well the purpose of this research since we can analyze how different document groups behave in different systems settings. This also allows us to see whether a single module yields better summaries than the combination of all the 4 modules for the selected 25 groups of documents. We have selected the following combinations of modules (please recall that the scoring module is the essential final step and thus common to all of them, so it is omitted from the description of combinations):

- **ASW** basic stopwords filtering
- **AR** pronominal anaphora are substituted by their antecedents prior to scoring
- **TE** redundant sentences are eliminated using textual entailment
- **TEWSD** members of the same WordNet[3] synset are replaced by the same synset representative and after that the redundant sentences are identified using the textual entailment module
- **ARTEWSD** pronominal anaphora are substituted by their antecedents, then the words of the resulting text are replaced by the chosen representative of the WordNet synset that they belong to and finally the redundant sentences are filtered out by the textual entailment module

Using these combination we generated summaries for all the 25 groups. Thereafter these summaries were evaluated using the ROUGE toolkit [14]. The average results group-wise are presented in Table 2.

---

[3] http://wordnet.princeton.edu/

Table 2. ROUGE-1 recall topicwise for generated summaries

		ASW	AR	TE	TEWSD	ARTE WSD
accidents	1. battleship explosion	0.41640	0.43224	0.41762	0.40645	0.42407
	2. ferry accidents	0.40627	0.42393	0.38874	0.38874	0.41932
	3. IRA attack	0.34765	0.38686	0.34955	0.34714	0.39822
	average	0.39010	0.41434	0.38530	0.38077	0.41387
natural disasters	4. earthquake Iran	0.39851	0.40446	0.42258	0.42936	0.42390
	5. China flood	0.44805	0.46006	0.48130	0.47697	0.46641
	6. Hurricane Gilbert	0.40462	0.41458	0.40446	0.40584	0.44845
	7. Mount Pinatubo volcano	0.32098	0.38782	0.31535	0.31535	0.36166
	8. North American drought	0.42936	0.44494	0.40973	0.41773	0.41594
	9. thunderstorm US	0.32589	0.37270	0.35346	0.36224	0.40338
	average	0.38790	0.41409	0.39781	0.40124	0.41995
politics	10. Checkpoint Charlie	0.50284	0.42880	0.46839	0.47730	0.50036
	11. abortion law	0.30283	0.38178	0.31252	0.31252	0.37216
	12. Germany reunification	0.40949	0.42933	0.40967	0.40868	0.45277
	13. Honecker protest	0.48695	0.48410	0.50062	0.49798	0.50517
	14. Iraq invades Kuwait	0.38630	0.39488	0.40467	0.40404	0.41887
	15. Robert Maxwell companies	0.42572	0.44064	0.42153	0.42411	0.44893
	16. striking coal miners	0.46955	0.44690	0.49049	0.49049	0.49765
	17. US ambassadors	0.45776	0.39415	0.47221	0.47242	0.45394
	average	0.43018	0.42507	0.43501	0.43594	0.45623
sports	18. Super Bowl	0.50047	0.45308	0.53126	0.53318	0.51541
	19. marathon	0.36541	0.35290	0.34855	0.34932	0.37978
	20. Olympics	0.42524	0.42857	0.47455	0.48249	0.48283
	average	0.43037	0.41151	0.45145	0.45499	0.45934
famous people	21. Leonard Bernstein	0.42669	0.42064	0.43319	0.43319	0.42277
	22. Lucille Ball	0.37099	0.36333	0.38172	0.37673	0.38633
	23. Margaret Thatcher	0.41783	0.42240	0.41403	0.43089	0.40990
	24. Sam Walton	0.36775	0.37903	0.35351	0.35351	0.40549
	25. Gorbachev	0.36199	0.37621	0.35525	0.35883	0.40405
	average	0.38905	0.39232	0.38754	0.39063	0.405708

5.2   Discussion

We pursued two different goals in this research. The initial hypothesis was that
the topic of the original document and the quality of generated summary are
related. The second one was that the quality of a generated summary rather
depends on linguistic properties of the original text and how they interact with
the particular summarization technique chosen to tackle this task.

Below is the analysis of the results with respect to both of the goals.

Does the Topic of the Original Document affect the Quality of Generated Summaries? On hand of the obtained results we couldn't prove the first
hypothesis. If we for example consider the *sports* topic, it becomes evident that:

– already the starting values for the ASW setting range from 0.36541 to
   0.50047

- the degree of improvement when adding additional modules differs between the three topics. For the *marathon* topic we started with the value of 0.36541 for the ASW setting and reached the maximum of 0.37978 for the ARTEWSD setting. Meanwhile for the *Olympics* the initial ROUGE value was 0.42524 and jumped up to 0.48283
- different modules and their combinations affected the quality of generated summaries in different ways. AR decreased the quality of summaries for the documents on *Super Bowl*, while TE and TEWSD setting noticeably improved it. The combination of both in the ARTEWSD setting yielded worse results than the mere TE and TEWSD settings. On the other hand, while the summaries of the documents on *Olympics* reveal the same tendencies for AR and TE/TEWSD settings, the combination of modules in ARTEWSD setting yielded the best results

No clear relation between the topic of a document and the summarization method performing best for it could be established. The same tendencies were observed for the remaining 4 general topics. Thus we can conclude that on this coarse-grained level of topic differentiation the performance of a summarization system does not depend on the topic of a document.

Do the Linguistic Properties of the Original Document affect the Quality of Generated Summaries? If we consider again the topic *Super Bowl* and the *Mount Pinatubo volcano*, we can see that the best results were obtained not for the combination of all the modules, but for the TE/TEWSD in the former case and the AR in the latter. The motivation for second objective of this research was to identify certain properties in the original text prior to summarizing to further chose the best summarization technique that will allow us to maximilly improve the quality of generated summaries. Though there are some exception, we could identify the following trends.

Textual Entailment. Whenever the noun ratio is above 0.35, textual entailment benefits the process of summarization (see for example the topics 4, 5, and 9 in Table 2). In the opposite case it worsens the results (as in topics 19, 24 or 25)

Anaphora Resolution. The chosen anaphora resolution tool seems to interact with the pronoun and proper noun ratios. It benefits when a) both values are low (e.g. topics 1, 2, 5, 7); b) proper noun ratio is low and pronoun ratio is high (as in topics 3, 11, 24); c) proper noun ratio is high and pronoun ratio is low (e.g. 4, 6, 12). The final thresholds for *high* and *low* ratios must be determined statistically with the larger corpus. In our case by *high* we mean "above the average" (thus highlighted in bold in Table 1) and *low* otherwise. When both ratios are high (topics 10 and 21), the anaphora resolution module tend to reduce the quality of generated summaries.

Textual Entailment and Anaphora Resolution. The interaction of textual entailment with anaphora resolution is less straightforward. In some cases when one should benefit and the other worsen the results, their combination still tops the results from the best performing module in isolation. For example, for the topic 25 AR improves the results of ASW. Both TE and TEWSD make them worse.

But the combination of AR and TEWSD improves over the results yielded on AR. The opposite case is observed for the topic 1, when AR benefits, TE yields worse results, and their combination is still worse than the results of AR only. We thus assume that there are more linguistic properties and ways of interaction with the summarization techniques that are the subject to further research.

Nevertheless, it becomes clear that the linguistic properties of the original document affect the quality of generated summaries.

6   Conclusion and Future Work

We have pursued two objectives in this research: i) determine whether the performance of a summarization system depends on the topic of a document; ii) determine whether the quality of a generated summary depends on the linguistic properties of the original text and how they interact with different summarization techniques. Given the findings discussed in Section 5.2, we conclude that i) no clear relation between the topic of a document and the can be established; ii) a preliminary document analysis stage could benefit the summarization process. The latter is valid both for modular summarization systems as described in Section 3 and any other summarization system based on a single method (i.e. latent semantic analysis or graph-based approaches), provided that their developers are aware of the inherent properties of the text their system can handle the best. Therefore it becomes advisable to adapt summarization systems to the linguistic properties of input documents.

To the best of our knowledge there has been no other study of the impact of linguistic properties of the original text on the quality of generated summary. Thus this research focused on a few linguistic properties, such as noun, pronoun and proper noun ratio and their interaction with the summarization heuristics involving textual entailment and anaphora resolution. For the future work we are planning to include other structural and semantic text properties. It is worth investigating how the word ambiguity index of the original document affects summarization systems that use word sense disambiguation and textual entailment. Another direction would lead to the structural features that include phrase type ratio, instead of mere noun or pronoun ratio, and parse tree analysis, that can be combined with text simplification for automatic summarization.

References

1. Afantenos, S., Karkaletsis, V., Stamatopoulos, P.: Summarization from medical documents: a survey. Artificial Intelligence in Medicine 33, 157–177 (2005)
2. Amini, M.-R., Gallinari, P.: The Use of Unlabeled Data to Improve Supervised Learning for Text Summarization. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2002, p. 105. ACM Press, New York (2002)
3. Ceylan, H., Mihalcea, R., Öyertem, U., Lloret, E., Palomar, M.: Quantifying the Limits and Success of Extractive Summarization Systems Across Domains. In: Human Language Technologies, pp. 903–911. Association for Computational Linguistics, Stroudsburg (2010)
4. Chuang, W.T., Yang, J.: Text Summarization by Sentence Segment Extraction Using Machine Learning Algorithms. In: Terano, T., Liu, H., Chen, A.L.P. (eds.) PAKDD 2000. LNCS, vol. 1805, pp. 454–457. Springer, Heidelberg (2000)
5. Edmunson, H.: New methods in automatic extracting. Journal of the ACM 16(2), 264–285 (1969)
6. Elhadad, N., McKeown, K., Kaufman, D., Jordan, D.: Facilitating physicians access to information via tailored text summarization. In: AMIA Annual Symposium, pp. 226–230 (2005)
7. Elhadad, N., Kan, M.-Y., Klavans, J.L., McKeown, K.R.: Customization in a Unified Framework for Summarizing Medical Literature. In: Artificial Intelligence in Medicine, vol. 33, pp. 179–198 (2005)
8. Filippova, K., Mieskes, M., Nastase, V.: Cascaded Filtering for Topic-Driven Multi-Document Summarization. In: Proceedings of the Document Understanding Conference, Rochester, N.Y., pp. 30–35 (2007)
9. Galley, M.: Automatic Summarization of Conversational Multi-Party Speech. In: The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, pp. 1914–1915. AAAI Press, Boston (2006)
10. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001), pp. 19–25. ACM Press, New York (2001)
11. Hu, M., Sun, A., Lim, E.: Comments-Oriented Blog Summarization by Sentence. In: Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management, pp. 901–904. Association for Computational Linguistics, New York (2007)
12. Kazantseva, A.: Automatic Summarization of Short Fiction, Master thesis (2006), http://www.site.uottawa.ca/ ankazant/pubs/ thesis_corrected_18_12_06_OK.pdf
13. Lee, D.: Genres, registers, text types, domains and styles: clarifying the concepts and navigating a path through the BNC jungle. Language and Computers 5, 37–72 (2002)
14. Lin, C.-Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proceedings of the Workshop on Text Summarization, p. 89 (2004)
15. Lloret, E., Ferrández, O., Muñoz, R., Palomar, M.: A Text Summarization Approach Under the Influence of Textual Entailment. In: 5th International Workshop on NLPCS, pp. 22–31 (2008)

16. Lloret, L., Palomar, M.: A Gradual Combination of Features for Building Automatic Summarisation Systems. In: Proceedings of the 12th International Conference on Text, Speech and Dialogue (TSD), Pilsen, Czech Republic, pp. 16–23 (2009)
17. Luhn, H.P.: The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development 2(2), 157–165 (1958)
18. McKeown, K., Hirschberg, J., Galley, M., Maskey, S.: From Text to Speech Summarization. In: International Conference on Acoustics, Speech, and Signal Processing, pp. 997–1000. IEEE, Philadelphia (2005)
19. Mihalcea, R., Ceylan, H.: Explorations in Automatic Book Summarization. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 380–389 (2007)
20. Muresan, S., Tzoukermann, E., Klavans, J.L.: Combining Linguistic and Machine Learning Techniques for Email Summarization. In: Proceedings of the 2001 Workshop on Computational Natural Language Learning (ConLL 2001). Association for Computational Linguistics, Stroudsburg (2001)
21. Nenkova, A., Chae, J., Louis, A., Pitler, E.: Empirical Methods in Natural Language Generation. Springer, Heidelberg (2010)
22. Nenkova, A.: Automatic Summarization. Foundations and Trends in Information Retrieval 5, 103–233 (2011)
23. Nenkova, A., Bagga, A.: Facilitating Email Thread Access by Extractive Summary Generation. In: Nicolov, N., Bontcheva, K., Angelova, G., Mitkov, R. (eds.) Recent Advances in Natural Language Processing III, Selected Papers from RANLP 2003, pp. 287–296. John Benjamins, Amsterdam (2003)
24. Plaza, L., Díaz, A.: Using Semantic Graphs and Word Sense Disambiguation. Techniques to Improve Text Summarization. Procesamiento del Lenguaje Natural 47, 97–105 (2011)
25. Saggion, H.: Topic-based Summarization at DUC 2005. In: Proceedings of the Document Understanding Workshop, Vancouver, B.C., Canada, pp. 1–6 (2005)
26. Steinberger, J., Poesio, M., Kabadjov, M.A., Ježek, K.: Two Uses of Anaphora Resolution in Summarization. Information Processing and Management 43(6), 1663–1680 (2007)
27. Tatar, D., Tamaianu-Morita, E., Mihis, A., Lupsa, D.: Summarization by Logic Segmentation and Text Entailment. In: 33rd CICLing, pp. 15–26 (2008)
28. Teufel, S., Moens, M.: Sentence extraction as a classification task. In: ACL/EACL 1997 Workshop on Intelligent Scalable Text Summarization, pp. 58–65. Association for Computational Linguistics, Madrid (1997)
29. Vodolazova, T., Lloret, E., Muñoz, R., Palomar, M.: A Comparative Study of the Impact of Statistical and Semantic Features in the Framework of Extractive Text Summarization. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2012. LNCS, vol. 7499, pp. 306–313. Springer, Heidelberg (2012)
30. Yang, J., Cohen, A.M., Hersh, W.: Automatic summarization of mouse gene information by clustering and sentence extraction from MEDLINE abstracts. In: AMIA Annual Symposium, pp. 831–835 (2007)

Unsupervised Medical Subject Heading Assignment Using Output Label Co-occurrence Statistics and Semantic Predications

Ramakanth Kavuluru[1,2,⋆] and Zhenghao He[2]

[1] Division of Biomedical Informatics, Department of Biostatistics
[2] Department of Computer Science
University of Kentucky, Lexington, KY
`{ramakanth.kavuluru,zhenghao.he}@uky.edu`

Abstract. Librarians at the National Library of Medicine tag each biomedical abstract to be indexed by their Pubmed information system with terms from the Medical Subject Headings (MeSH) terminology. The MeSH terminology has over 26,000 terms and indexers look at each article's full text to assign a set of most suitable terms for indexing it. Several recent automated attempts focused on using the article title and abstract text to identify MeSH terms for the corresponding article. Most of these approaches used supervised machine learning techniques that use already indexed articles and the corresponding MeSH terms. In this paper, we present a novel unsupervised approach using named entity recognition, relationship extraction, and output label co-occurrence frequencies of MeSH term pairs from the existing set of 22 million articles already indexed with MeSH terms by librarians at NLM. The main goal of our study is to gauge the potential of output label co-occurrence statistics and relationships extracted from free text in unsupervised indexing approaches. Especially, in biomedical domains, output label co-occurrences are generally easier to obtain than training data involving document and label set pairs owing to the sensitive nature of textual documents containing protected health information. Our methods achieve a micro F-score that is comparable to those obtained using supervised machine learning techniques with training data consisting of document label set pairs. Baseline comparisons reveal strong prospects for further research in exploiting label co-occurrences and relationships extracted from free text in recommending terms for indexing biomedical articles.

1 Introduction

Indexing biomedical articles is an important task that has a significant impact on how researchers search and retrieve relevant information. This is especially essential given the exponential growth of biomedical articles indexed by PubMed®, the main search system developed by the National Center for Biotechnology

---

⋆ Corresponding author.

Information (NCBI). PubMed lets users search over 22 million biomedical citations available in the MEDLINE bibliographic database curated by the National Library of Medicine (NLM) from over 5000 leading biomedical journals in the world. To keep up with the explosion of information on various topics, users depend on search tasks involving Medical Subject Headings (MeSH®) that are assigned to each biomedical article. MeSH is a controlled hierarchical vocabulary of medical subjects created by the NLM. Once articles are indexed with MeSH terms, users can quickly search for articles that pertain to a specific subject of interest instead of relying solely on key words based searches.

Since MeSH terms are assigned by librarians who look at the full text of an article, they capture the semantic content of an article that cannot easily be captured by key word or phrase searches. Thus assigning MeSH terms to articles is a routine task for the indexing staff at NLM. This is empirically shown to be a complex task with 48% consistency because it heavily relies on indexers' understanding of the article and their familiarity with the MeSH vocabulary [1]. As such, the manual indexing task takes a significant amount of time leading to delays in the availability of indexed articles. It is is observed that it takes about 90 days to complete 75% of the citation assignment for new articles [2]. Moreover, manual indexing is also a fiscally expensive initiative [3]. Due to these reasons, there have been many recent efforts to come up with automatic ways of assigning MeSH terms for indexing biomedical articles. However, automated efforts (including ours) mostly focused on predicting MeSH terms for indexing based solely on the abstract and title text of the articles. This is because most full text articles are only available based on paid licenses not subscribed by many researchers.

Many efforts in MeSH term prediction generally rely on two different methods. The first method is the k-nearest neighbor (k-NN) approach where k articles that are already tagged with MeSH terms and whose content is found to be "close" to the new abstract to be indexed are obtained. The MeSH terms from these k articles form a set of candidate terms for the new abstract. A second method is based on applying machine learning algorithms to learn binary classifiers for each MeSH term. A new candidate abstract would then be put through all the classifiers and the corresponding MeSH terms of classifiers that return a positive response are chosen as the indexed terms for the abstract. We note that both k-NN and machine learning approaches need large sets of abstracts and the corresponding MeSH terms to make predictions for new abstracts. In this paper, we propose an unsupervised ensemble approach to extract MeSH terms and test it on two gold standard datasets. Our approach is based on named entity recognition (NER), relationship extraction, knowledge-based graph mining, and output label co-occurrence statistics. Prior attempts have used NER and graph mining approaches as part of their supervised approaches and we believe this is the first time relationship extraction and output label co-occurrences are applied for MeSH term extraction. Furthermore, our approach is purely unsupervised in that we do not use a prior set of already tagged MEDLINE citations with their corresponding MeSH terms.

Before we continue, we would like to emphasize that automatic indexing attempts, including our current attempt, are generally not intended to replace

trained indexers but are mainly motivated to expedite the indexing process and increase the productivity of the indexing initiatives at the NLM. Hence in these cases, recall might be more important than precision although an acceptable trade-off is necessary. In the rest of the paper, we first discuss related work and the context of our paper in Section 2. We describe our dataset and methods in Section 3. We provide an overview of the evaluation measures and present results with discussion in Section 4.

2   Related Work

NLM initiated efforts in MeSH term extraction with their Medical Text Indexer (MTI) program that uses a combination of k-NN based approach and NER based approaches with other unsupervised clustering and ranking heuristics in a pipeline [4]. MTI recommends MeSH terms for NLM indexers to assist in their efforts to expedite the indexing process[1]. Another recent approach by Huang et al. [2] uses k-NN approach to obtain MeSH terms from a set of k already tagged abstracts and use the *learning to rank* approach to carefully rank the MeSH terms. They use two different gold standard datasets one with 200 abstracts and the other with 1000 abstracts. They achieve an F-score of 0.5 and recall 0.7 on the smaller dataset compared to MTI's F-score of 0.4 and recall 0.57. Several other attempts have tried different machine learning approaches with novel feature selection [5] and training data sample selection [6] techniques. A recent effort by Jimeno-Yepes et al. [7] uses a large dataset and uses meta-learning to train custom binary classifiers for each label and index the best performing model for each label for applying on new abstracts; we request the reader to refer to their work for a recent review of machine learning used for MeSH term assignment. As mentioned in Section 1, most current approaches rely on large amounts of training data. We take a purely unsupervised approach under the assumption that we have access to output label[2] co-occurrence frequencies where training documents may not be available.

3   Our Approach

We use two different datasets, a smaller 200 abstract dataset and a larger 1000 abstract dataset used by Huang et al. [2]; besides results from their approach, they also report on the results produced by NLM's MTI system. We chose these datasets and compare our results with their outcomes as they represent the k-NN and machine learning approaches typically used by most researchers to address MeSH term extraction. To extract MeSH terms, we used a combination of three methods: NER, knowledge-based graph mining, and output label co-occurrence

---

[1] For the full architecture of MTI's processing flow, please see: `http://skr.nlm.nih.gov/resource/Medical_Text_Indexer_Processing_Flow.pdf`

[2] Here the 'labels' are MeSH terms; we use 'label' to conform to the notion of classes in multi-label classification problems.

statistics to extract candidate MeSH terms. We finally use semantic predications to rank the candidates and also use the traditional Borda rank aggregation method to rank various ranked lists of the candidate set. In this section we elaborate on the specifics of each of these components of our approach. Before we proceed, we first discuss the Unified Medical Language System (UMLS), a biomedical knowledge base used in NER, graph mining methods, and extraction of semantic predications.

3.1 Unified Medical Language System (UMLS)

The UMLS[3] is a large domain expert driven aggregation of over 160 biomedical terminologies and standards. It functions as a comprehensive knowledge base and facilitates interoperability between information systems that deal with biomedical terms. It has has three main components: Metathesaurus, Semantic Network, and SPECIALIST lexicon. The Metathesaurus has terms and codes, henceforth called *concepts*, from different terminologies. Biomedical terms from different vocabularies that are deemed synonymous by domain experts are mapped to the same Concept Unique Identifier (CUI) in the Metathesaurus. The semantic network acts as a typing system that is organized as a hierarchy with 133 *semantic types* such as *disease or syndrome, pharmacologic substance,* or *diagnostic procedure.* It also captures 54 important relations (called semantic relations) between biomedical entities in the form of a relation hierarchy with relations such as *treats, causes,* and *indicates.* The Metathesaurus currently has about 2.8 million concepts with more than 12 million relationships connecting these concepts. The relationships take the form $C1 \rightarrow <rel-type> \rightarrow C2$ where $C1$ and $C2$ are concepts in the UMLS and $<rel-type>$ is a semantic relation such as treats, causes, or interacts. The semantic interpretation of these relationships (also called triples) is that the $C1$ is related to $C2$ via the relation $<rel-type>$. The SPECIALIST lexicon is useful for lexical processing and variant generation of different biomedical terms.

3.2 Named Entity Recognition: MetaMap

NER is a well known application of natural language processing (NLP) techniques where different entities of interest such as people, locations, and institutions are automatically recognized from mentions in free text (see [8] for a survey). Named entity recognition in biomedical text is difficult because linguistic features that are normally useful (e.g., upper case first letter, prepositions before an entity) in identifying generic named entities are not useful when identifying biomedical named entities, several of which are not proper nouns. Hence, NER systems in biomedicine rely on expert curated lexicons and thesauri. In this work, we use MetaMap [9], a biomedical NER system developed by researchers at the National Library of Medicine (NLM). So as the first step in

---

[3] UMLS Reference Manual: `http://www.ncbi.nlm.nih.gov/books/NBK9676/`

identifying MeSH terms for a given abstract, we extract non-negated biomedical named entities by running MetaMap on the abstract text using MetaMap's ability to identify negated terms. Once we obtain non-negated UMLS concepts using MetaMap from the abstract text, we convert these concepts to MeSH terms, when possible. Specifically, we first note that MeSH is one of the over 160 source vocabularies integrated into the UMLS Metathesaurus. As such, concepts in MeSH also have a concept unique identifier (CUI) in the Metathesaurus. As part of its output, for each concept, MetaMap also gives the source vocabulary. The concepts from MetaMap with source vocabulary MeSH finally become the set of extracted 'candidate' terms for each abstract. However, these MeSH term sets may not be complete because of missing relationships between UMLS concepts. That is, in our experience, although MetaMap identifies a medical subject heading, it might not always map it to a CUI associated with a MeSH term; it might map it to some other terminology different from MeSH, in which case we miss a potential MeSH term because the UMLS mapping is incomplete. We deal with this problem and explore a graph based approach in the next section. We also note that just because a MeSH term appears in the abstract, it may not be the case that the abstract should be tagged with that term (more on this later).

3.3   UMLS Knowledge-Based Graph Mining

As discussed in Section 3.2, the NER approach might result in poor recall because of lack of completeness in capturing synonymy in the UMLS. However, using the UMLS graph with CUIs as nodes and the inter-concept relationships connected by relationship types *parent* and *rel_broad* as edges (high level relationship types in UMLS), we can map a original CUI without an associated MeSH term to a CUI with an associated MeSH term. The *parent* relationship means that concept $C1$ has $C2$ as a *parent*. The *rel_broad* type means that $C1$ represents a broader concept than $C2$. We adapt the approach originally proposed by Bodenreider et al. [10] for this purpose. The mapping algorithm starts with a CUI c output by MetaMap that is not associated with an MeSH term and tries to map it to an MeSH term as follows.

1. Recursively, construct a subgraph G_c (of the UMLS graph) consisting of ancestors of the input non-MeSH CUI c, using the *parent* and *rel_broad* edges. Build a set I_c of all the MeSH concepts associated with nodes added to G_c along the way in the process of building G_c. Note that many nodes added to G_c may not have associated MeSH terms.
2. Delete any concept c_1 from I_c if there exists another concept c_2 such that
   - c_1 is an ancestor of c_2, and
   - The length of the shortest path from c to c_2 is less than the length of the shortest path from c to c_1.
3. Return the MeSH terms of remaining concepts in I_c and the corresponding shortest distances from c.

Note the the algorithm essentially captures ancestors of the input concept and tries to find MeSH headings in them.

3.4   Candidate Set Expansion Using Output Label Co-Occurrences

Using NER and graph-based mining discussed in Sections 3.2 and 3.3, we obtain a pool of candidate MeSH terms. However, note that the trained coders will look at the entire full text to assign MeSH terms to the articles. Thus, merely looking for MeSH terms mentioned in the title or the abstract may not be sufficient. To further expand the pool of MeSH candidates we propose to exploit the frequencies of term co-occurrences as noticed in already indexed articles. To elaborate, we already have nearly 22 million articles that are manually assigned MeSH terms from which we can count the number of times different term pairs co-occur in the form a matrix where both rows and columns are all possible MeSH terms (nearly 26,000). Before we go into specific details, we give a high level overview of our approach to exploit output term co-occurrences. Intuitively, given a MeSH term that *we already know with high confidence should be assigned to a particular abstract*, other terms that frequently co-occur with the known term might also make good candidates for the input abstract. However,

1. there might be many highly co-occurrent terms; high co-occurrence does not necessarily mean that the new term is relevant in the context of the current abstract that is being assigned MeSH terms. To address this, we propose to model the *context* using MeSH terms extracted from title and abstract using NER and graph-mining (Sections 3.2 and 3.3). We still need a way of *applying* this context to separate highly co-occurrent terms that are also relevant for the current abstract.
2. Furthermore, we also need an initial seed set of high confidence candidate terms to exploit the term co-occurrences. We propose to use, again, the MeSH terms extracted from title and abstract using NER and graph-mining. The title MeSH terms are directly included in the seed set of candidate terms. However, the terms extracted using NER from the abstract are subject to the context (as indicated in the first step in this list) and are only included in the seed set if they are still deemed relevant after applying the context[4].

Given the outline explained thus far, next we present specifics of how the highly co-occurring terms are obtained from the seed set and how the context terms (that is, MeSH terms from title and abstract) are used to select a few highly co-occurrent terms that are also contextually relevant for the current article to be indexed. Before we proceed, as a pre-processing step, we build a two dimensional matrix $\mathcal{M}$[5] of row-normalized term co-occurrence frequencies where both rows and columns are all possible MeSH terms and the cells are defined as

$$\mathcal{M}[i][j] = \frac{\text{number of articles assigned both } i\text{-th and } j\text{-th MeSH terms}}{\text{number of articles assigned the } i\text{-th term}}.$$

---

[4] This is needed because MeSH terms that are mentioned in the abstract may not be relevant to the article. An example situation is when a list of diseases is mentioned in the abstract although the article is not about any of them but about the biology of a particular protein that was implicated in all those diseases.

[5] We used the Compressed Sparse Row matrix class from the `SciPy` Python package to efficiently represent and access the 26000×26000 matrix.

Here $\mathcal{M}[i][i] = 1$ because the numerator is just the same as the denominator. We note with this definition of $\mathcal{M}[i][j]$ is an estimate of the probability $P(j\text{-th term}|i\text{-th term})$. Let $\mathcal{T}$ and $\mathcal{A}$ be the set of title and abstract MeSH terms extracted using NER, respectively, and $\mathcal{C} = \mathcal{T} \cup \mathcal{A}$ be the set of context terms which includes the MeSH terms extracted from both title and abstract. Let α and β be the thresholds used to identify highly co-occurrent terms and to select a few of these terms that are also contextually relevant, respectively. Details of these thresholds will be made clear later in this section. Next we show the pseudocode of candidate term expansion algorithm.

Algorithm. `Expand-Candidate-Terms` $(\mathcal{T}, \mathcal{A}, \alpha, \beta, \mathcal{M}[][])$

1: Initialize seed list $S = \mathcal{T}$
2: Set context terms $\mathcal{C} = \mathcal{T} \cup \mathcal{A}$
3: $S.append(\texttt{Apply-Context}(\mathcal{A}, \beta, \mathcal{C}, \mathcal{M}[][]))$
   {Next, we iterate over terms in list S}
4: **for all** terms t in S **do**
5:     Let $H = []$ be an empty list
6:     **for** each i such that $\mathcal{M}[t][i] > \alpha$ **do**
7:         $H.append(i\text{-th MeSH term})$
8:     $relevantTerms = \texttt{Apply-Context}(H, \beta, \mathcal{C}, \mathcal{M}[][])$
9:     $relevantTerms = relevantTerms - S$ {avoid adding existing terms}
10:     $S.append(relevantTerms)$
11: return S

Procedure. `Apply-Context` $(H, \beta, \mathcal{C}, \mathcal{M}[][])$

1: **for all** candidate terms t in H **do**
2:     Set co-occurrence score $F = 0$
3:     **for** each context term c in $\mathcal{C}$ **do**
4:         $F = F + \mathcal{M}[c][t]$
5:     **if** $F/|\mathcal{C}| < \beta$ **then**
6:         $H.delete(t)$ {$F/|\mathcal{C}|$ is the average co-occurrence}
7: return H

Next, we discuss the `Expand-Candidate-Terms` algorithm. It takes the title and abstract MeSH terms as input and also the thresholds α, to extract the highly co-occurring terms with the seed terms, and β to apply context and prune the expanded set of terms. We initialize the seed set to be just the title terms (line 1). In line 3, we add to the seed set, abstract terms that have an average co-occurrence score $\geq \beta$ with the context terms. In lines 4–10, we expand the seed set to add new candidate terms. For each seed term t considered in the `for` loop on line 4, we curate a list of highly co-occurring terms according to the term pair co-occurrence matrix (lines 6–7). We then prune this list of terms based on their average co-occurrence with context terms by calling `Apply-Context` in line 8. To ensure termination and avoid looking at terms that we have already expanded, we only append terms that are not already in S (lines 9–10).

In the `Apply-Context` procedure, we add the co-occurrence scores of each term in the list H with all terms in the context term set $\mathcal{C}$ (lines 3–4). We delete all terms from H that have an average co-occurrence less than β. In our experiments, $0.03 \leq \beta \leq 0.05$ and $0.06 \leq \alpha \leq 0.1$ proved to be best ranges for the thresholds. Using very low thresholds will increase the size of the expanded candidate set output by `Expand-Candidate-Terms` (line 11). Given this expanded candidate set, we rank its terms to retain only a top few; in our experiments, the candidate sets were found to have anywhere between 25 to 200 terms while the label cardinality of our datasets is only close to 15.

3.5   Ranking Approaches and Semantic Predications

In this section, we explore different unsupervised ranking approaches to rank the resulting candidate MeSH terms obtained using the methods from Section 3.4. A straightforward method we use is to rank them based on the average co-occurrence score computed in line 5 ($F/|\mathcal{C}|$) of the procedure `Apply-Context` from Section 3.4; a second approach we follow is to to rank by the number of context terms in $\mathcal{C}$ with which the candidate term has a co-occurrence value $\geq$ the average co-occurrence on line 5. That is the number of terms c such that $\mathcal{M}[c][t] \geq F/|\mathcal{C}|$ in `Apply-Context`. Both these approaches are based on our co-occurrence frequency based methods.

We also experiment with a novel binning approach using binary relationships (popularly called *semantic predications*) extracted from the abstract text using the SemRep, a relationship extraction program developed by Thomas Rindflesch [11] and team at the NLM. Semantic predications are of the form $C1 \rightarrow <rel - type> \rightarrow C2$ discussed in Section 3.1. However, the relationships come from the abstract text instead of the UMLS source vocabularies. The intuition is that entities $C1$ and $C2$ that participate as components of binary relationships should be ranked higher than those that do not participate in any such relationship. By virtue of participating in a binary relationship asserted in one of the sentences of the abstract text, we believe they garner more importance as opposed to just being mentioned in a list of things in the introductory sentences of an abstract. Thus we divide the set of candidate terms from Section 3.4 into two bins. The first bin contains those MeSH terms that participate as a subject or an object of a semantic predication extracted from the text. The second bin consists of those candidate terms that did not occur as either a subject or an object of some predication. Terms in the first bin are always ranked higher than terms in the second bin. Within each bin, terms are ranked according to their average co-occurrence score or according to the number of context terms with which the candidate term has co-occurrence $\geq$ the average. We also subdivided each main bin into two sub-bins where the first sub-bin consists of those terms that are extracted from the abstract (using NER) and the second that consists of only those terms that were extracted using the co-occurrence statistics. Again, ranking within sub-bins is based on scores resulting from the co-occurrence based expansion algorithms. Finally we used Borda's [12] positional rank aggregation method to aggregate different full rankings produced by

purely co-occurrence based scoring methods and bin-based scoring methods. In all these approaches, ties are broken using the average co-occurrence score and the rare ties where these scores are equal are broken by maintaining the original order in which terms are added in the expansion algorithm.

Remark 1. We also curate a small set of generic MeSH terms that lead to very large number of false positives (e.g., *Disease, Persons, Patients*), mostly generic terms (including some check-tags[6]) and then apply a discount to the scores of these terms if they are found in the candidate terms.

4   Experiments, Results, and Discussion

Before we discuss our findings, we establish the notation to be used for evaluation measures. Let D be the set of all biomedical abstracts to be tagged with MeSH terms; Let E_i and G_i, $i = 1, \ldots, |M|$, be the set of extracted MeSH terms using our methods from the PubMed citations (here, abstract and title fields) and the corresponding correct gold standard terms, respectively, for the i-th citation. Based on methods discussed in Section 3.5, we also impose a ranking on terms in B_i and only use the top N terms for computing performance measures. Since the task of assigning multiple terms to an abstract is the multi-label classification problem, there are multiple complementary methods for evaluating automatic approaches for this task. However, since we are using an unsupervised approach, we limit ourselves to the micro precision, recall, and F-score used by Huang et al [2]. The average micro precision P_μ and recall R_μ are

$$P_\mu = \frac{\sum_{i=1}^{|D|} c(N, D_i, E_i)}{|D| \cdot N} \quad \text{and} \quad R_\mu = \frac{\sum_{i=1}^{|D|} c(N, D_i, E_i)}{\sum_{i=1}^{|M|} |G_i|},$$

where $c(N, D_i, E_i)$ is the number of true positives (correct gold standard terms) in the top N ranked list of candidate terms in E_i for citation D_i. Given this, the micro F-score is $F_\mu = 2P_\mu R_\mu / (P_\mu + R_\mu)$. We also define average precision of a citation $AP(D_i)$ computed considering top N terms as

$$AP(D_i, N) = \frac{1}{|G_i|} \sum_{r=1}^{N} I(E_i^r) \cdot \frac{c(r, D_i, E_i)}{r},$$

where E_i^r is the r-th ranked term in the set of predicted terms E_i for citation D_i and the function $I(E_i^r)$ is a Boolean function with a value of 1 if $E_i^r \in G_i$ and 0 otherwise. Finally, the mean average precision (MAP) of the collection of citations D when considering top N predicted terms is given by

$$MAP(D, N) = \frac{1}{|D|} \sum_{i=1}^{|D|} AP(D_i, N).$$

---

[6] Check-tags form a special small set of MeSH terms that are always checked by trained coders for all articles. Here is the full check tag list: `http://www.nlm.nih.gov/bsd/indexing/training/CHK_010.htm`

Remark 2. In our experiments, MeSH terms that are associated with concepts at a distance greater than 1 from the input concept in the graph mining approach (Section 3.3) did not provide a significant improvement in the results. Hence here we only report results when the shortest distance between the input concept and the MeSH ancestors is ≤ 1.

We used two different datasets – the smaller dataset has 200 citations and is called the NLM2007 dataset. The larger 1000 citation dataset is denoted by L1000. Both datasets can be obtained from the NLM website: `http://www.ncbi.nlm.nih.gov/CBBresearch/Lu/indexing/paperdat.zip`. Next, we present our best micro average precision, recall, F-score, and MAP in Table 1 in comparison with the results obtained by supervised ranking method by [2] and the results obtained when using NLM's MTI program (as reported by Huang et al. in their paper). From the table we see that the performance of our unsupervised methods is comparable (except in the case of the MAP measure) to that of the MTI method, which uses a k-NN approach. However, as can be seen, a supervised ranking approach that relies on training data and uses the k-NN approach performs much better than our approaches. We emphasize that our primary goal has been to demonstrate the potential of unsupervised approaches that can complement supervised approaches when training data is available but can work with reasonable performance even when training data is scarce or unavailable, which is often the case in many biomedical applications. Furthermore, unlike in many unsupervised scenarios, we do not even have access to the full artifact (here, full text of the article) to be classified, which further demonstrates the effectiveness of our method.

Table 1. Comparison of micro measures with $N = 25$

	NLM2007 dataset				L1000 dataset			
	R_μ	P_μ	F_μ	MAP	R_μ	P_μ	F_μ	MAP
Our method	0.54	0.32	0.40	0.36	0.56	0.29	0.38	0.38
MTI	0.57	0.31	0.40	0.45	0.58	0.30	0.39	0.46
Huang et al.	0.71	0.39	0.50	0.62	0.71	0.34	0.46	0.61

Next we contrast the performance of our unsupervised methods involving co-occurrence statistics and semantic predication based ranking approaches with some baseline methods that only use NER and graph-mining based approaches in Table 2; we do not show MAP values because the baseline approaches do not involve a ranking scheme. We see that graph-mining approach did not increase recall by more than 2%[7]. However, our co-occurrence based candidate term expansion (Section 3.4) improved the recall by 18% in both the NLM2007 and L1000 datasets with an increase in precision of at least 10% and an increase in

---

[7] We note that this is because we only used it for a specific set of qualifier terms that are in MeSH but needed a graph-based mapping to obtain the MeSH main headings.

Table 2. Comparison with baseline measures

	NLM2007 dataset			L1000 dataset		
	R_μ	P_μ	F_μ	R_μ	P_μ	F_μ
Our best scores	0.54	0.32	0.40	0.56	0.29	0.38
NER only	0.35	0.20	0.25	0.36	0.19	0.25
NER+graph-mining	0.36	0.19	0.25	0.38	0.18	0.24

F-score of at least 14%. This shows that using simplistic approaches that rely only on NER may not provide reasonable performance.

Whether using unsupervised or supervised approaches, fine tuning the parameters is always an important task. Next, we discuss how different thresholds (α and β in Section 3.4) and different values of N effect the performance measures. We believe this is important because low values for thresholds and high cut-off values for N have the potential to increase recall by trading-off some precision. We experimented with different threshold ranges for α and β and also different values of N. We show some interesting combinations we observed for the L1000 dataset in Table 3. We gained a recall of 1% by changing N from 25 to 35 with the same thresholds. Lowering the thresholds with $N = 35$ lead to a 5% gain in recall with an equivalent decrease in precision, which decreases the F-score by 5% while increasing the MAP score by 1%.

Table 3. Different combinations of N, α, and β

	L1000 dataset			
	R_μ	P_μ	F_μ	MAP
$N = 25, \alpha = 0.10, \beta = 0.05$	0.51	0.33	0.40	0.36
$N = 25, \alpha = 0.08, \beta = 0.04$	0.56	0.29	0.38	0.38
$N = 35, \alpha = 0.08, \beta = 0.04$	0.57	0.28	0.38	0.38
$N = 35, \alpha = 0.06, \beta = 0.03$	0.62	0.23	0.33	0.39

Finally, among the ranking approaches we tried, the best ranking method is Borda's aggregation of the two ranked lists, the first of which is based on average co-occurrence scores and the second is the semantic predication based binning approach with average co-occurrence as the tie-breaker within each bin. This aggregated ranking is used to obtain the best scores we reported in all the tables discussed in this section. The semantic predication based binning provided a 3% improvement in the MAP score both in the NLM2007 and L1000 datasets.

5 Conclusion

In this paper, we presented a novel unsupervised approach to assigning medical subject headings (MeSH terms) to biomedical articles. We deviate from the traditional k-NN approach and supervised machine learning approaches and use

named entity recognition, relationship extraction, and term pair co-occurrence statistics to perform a constrained expansion of a seed set of terms. We use semantic predications to bin candidate terms and then applied average co-occurrence scores (computed using normalized co-occurrence frequencies with certain context terms) to rank terms within the bins. We then used Borda's rank aggregation method to combine different ranked lists. Micro measures obtained using our methods are comparable to those obtained using k-NN based approaches such as the MTI program from NLM. More advanced learning-to-rank approaches did better than our methods. However, we believe our methods are an important contribution because they do not use any pre-labeled training data and are more suitable when training data is not available or is very limited, which can arise in biomedical and clinical domains. Furthermore, our methods can complement supervised approaches for labels with fewer training examples.

References

1. Funk, M., Reid, C.: Indexing consistency in medline. Bulletin of the Medical Library Association 71(2), 176 (1983)
2. Huang, M., Névéol, A., Lu, Z.: Recommending mesh terms for annotating biomedical articles. J. of the American Medical Informatics Association 18(5), 660–667 (2011)
3. Aronson, A., Bodenreider, O., Chang, H., Humphrey, S., Mork, J., Nelson, S., Rindflesch, T., Wilbur, W.: The nlm indexing initiative. In: Proceedings of the AMIA Symposium, American Medical Informatics Association, p. 17 (2000)
4. Aronson, A., Mork, J., Gay, C., Humphrey, S., Rogers, W.: The NLM indexing initiative: Mti medical text indexer. In: Proceedings of MEDINFO (2004)
5. Yetisgen-Yildiz, M., Pratt, W.: The effect of feature representation on medline document classification. In: AMIA Annual Symposium Proceedings, American Medical Informatics Association, vol. 2005, pp. 849–853 (2005)
6. Sohn, S., Kim, W., Comeau, D.C., Wilbur, W.J.: Optimal training sets for bayesian prediction of MeSH assignment. Journal of the American Medical Informatics Association 15(4), 546–553 (2008)
7. Jimeno-Yepes, A., Mork, J.G., Demner-Fushman, D., Aronson, A.R.: A one-size-fits-all indexing method does not exist: Automatic selection based on meta-learning. JCSE 6(2), 151–160 (2012)
8. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1), 3–26 (2007)
9. Aronson, A.R., Lang, F.M.: An overview of metamap: historical perspective and recent advances. J. American Medical Informatics Assoc. 17(3), 229–236 (2010)

10. Bodenreider, O., Nelson, S., Hole, W., Chang, H.: Beyond synonymy: exploiting the umls semantics in mapping vocabularies. In: Proceedings of AMIA Symposium, pp. 815–819 (1998)
11. Rindflesh, T.C., Fiszman, M.: The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. J. of Biomedical Informatics 36(6), 462–477 (2003)
12. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: Proceedings of the 10th International Conference on World Wide Web, WWW 2001, pp. 613–622 (2001)

Bayesian Model Averaging and Model Selection for Polarity Classification

Federico Alberto Pozzi, Elisabetta Fersini, and Enza Messina

University of Milano-Bicocca
Viale Sarca, 336 - 20126 Milan, Italy
`{federico.pozzi,fersini,messina}@disco.unimib.it`

Abstract. One of the most relevant task in Sentiment Analysis is Polarity Classification. In this paper, we discuss how to explore the potential of ensembles of classifiers and propose a voting mechanism based on Bayesian Model Averaging (BMA). An important issue to be addressed when using ensemble classification is the model selection strategy. In order to help in selecting the best ensemble composition, we propose an heuristic aimed at evaluating the a priori contribution of each model to the classification task. Experimental results on different datasets show that Bayesian Model Averaging, together with the proposed heuristic, outperforms traditional classification methods and the well known Majority Voting mechanism.

1 Introduction

According to the definition reported in [1], sentiment *"suggests a settled opinion reflective of one's feelings"*. The aim of Sentiment Analysis (SA) is therefore to define automatic tools able to extract subjective information, such as opinions and sentiments from texts in natural language, in order to create structured and actionable knowledge to be used by either a Decision Support System or a Decision Maker. The polarity classification task can be addressed at different granularity levels, such as word, sentence and document level. The most widely studied problem is SA at document level [2], in which the naive assumption is that each document expresses an overall sentiment. When this is not ensured, a lower granularity level of SA could be more useful and informative. In this work, polarity classification has been investigated at sentence level. The main polarity classification approaches are focused on identifying the most powerful model for classifying the polarity of a text source. However, an ensemble of different models could be less sensitive to noise and could provide a more accurate prediction [3]. Regarding SA, the study of ensembles is still on its infancy. This is mainly due to the difficulty to find out a reasonable trade-off between classification accuracy and increasing computational time, that is particularly challenging when dealing with online and real-time big data. To the best of our knowledge, the existing approaches of a voting system for SA are based on traditional methods such as Bagging [4] and Boosting [5], disregarding how to select the best ensemble composition. In this paper we propose a novel BMA approach that combines different models selected using a specific selection strategy heuristic.

2   Bayesian Model Averaging

The idea behind a voting mechanism is to exploit the characteristics of several independent classifiers by combining them in order to achieve better performance than the best single classifier. The most popular ensemble model is the Majority Voting (MV), which is characterized by an ensemble of "experts" that classifies the sentence polarity by considering the vote of each classifier as "equally important" and by determining the final polarity by selecting the most popular label prediction [3].

Let C be a set of n independent classifiers and $l_i(s)$ the label assigned to a sentence s by classifier $i \in C$. Then, the optimal label $l^{\mathrm{MV}}(s)$ is assigned as follows:

$$l^{\mathrm{MV}}(s) = \begin{cases} \text{positive if } \sum_{i \in C} l_i(s)_+ > \sum_{i \in C} l_i(s)_- \\ \text{negative if } \sum_{i \in C} l_i(s)_+ < \sum_{i \in C} l_i(s)_- \\ \widehat{l}(s) \qquad\quad otherwise \end{cases} \tag{1}$$

where $l_i(s)_+ = 1$ if the label assigned by i to s is positive (0 otherwise), $l_i(s)_- = 1$ if the label assigned by i to s is negative (0 otherwise) and $\widehat{l}(s)$ is the label assigned to s by the "most expert" classifier, i.e. the classifier that is able to ensure the highest accuracy.

A voting mechanism can be improved by explicitly taking into account the marginal distribution of each classifier prediction and its overall reliability when determining the optimal label. To this purpose, we propose a voting mechanism based on Bayesian Model Averaging (BMA) [6], where the weighted contribution of each classifier is used to make a final label prediction. This approach assigns to s the label $l^{\mathrm{BMA}}(s)$ that maximizes:

$$\begin{aligned} P(l(s)|C, \mathcal{D}) &= \sum_{i \in C} P(l(s)|i)P(i|\mathcal{D}) \\ &= \sum_{i \in C} P(l(s)|i)P(i)P(\mathcal{D}|i) \end{aligned} \tag{2}$$

where $P(l(s)|i)$ is the marginal distribution of the label predicted by classifier i, while $P(\mathcal{D}|i)$ represents the likelihood of the training data $\mathcal{D}$ given i. The prior $P(i)$ of each classifier is assumed to be a constant and therefore can be omitted. The distribution $P(\mathcal{D}|i)$ can be approximated by using the F_1-measure obtained during a preliminary evaluation of classifier i:

$$P(\mathcal{D}|i) \propto \frac{2 \times P_i(\mathcal{D}) \times R_i(\mathcal{D})}{P_i(\mathcal{D}) + R_i(\mathcal{D})} \tag{3}$$

where $P_i(\mathcal{D})$ and $R_i(\mathcal{D})$ denotes precision and recall obtained by classifier i.

According to (2), we take into account the vote of each classifier by exploiting the prediction marginal instead of a 0/1 vote and we tune this "probabilistic claim" according to the ability of the classifier to fit the training data. This approach allows the uncertainty of each classifier to be taken into account, avoiding over-confident inferences.

3 Model Selection Strategy

An important issue related to voting mechanisms is referred to the selection of models to be included in an ensemble. The best composition of classifiers is a combinatorial optimization problem over a dimension of $\sum_{k=1}^{n} \frac{n!}{k!(n-k)!}$ where n is the number of classifiers and k represents the dimension of each potential ensemble. For example, if we want to find out the best ensemble given 10 classifiers, we should test more than 1000 potential ensembles before determining the optimal one. In order to reduce the search space, we propose an heuristic able to compute the discriminative contribution that each classifier is able to provide with regard to other classifiers.

Given two classifiers, i and j, i could help j to globally tag a sentence with the correct label l when:

1. j incorrectly labels the sentence s, but i correctly tags it. This is the most important contribution of i to the voting mechanism and represents how much i is able to positively correct j;
2. Both i and j correctly label s. In this case, i enhances the weight used to choose the correct label.

On the other hand, i could also damage the ensemble in the following cases:

3. j correctly labels sentence s, but i incorrectly tags it. This is the most harmful contribution in a voting mechanism and represents how much i is able to negatively change the (correct) label tagged by j.
4. j incorrectly labels sentence s that has been misclassified also by i. In this case, i cooperates to further decrease the weight that the voting mechanism uses to chose the correct label.

To formally represents the cases above, we consider $P(i = 1|j = 0)$ as the number of instances correctly classified by i over the number of instances incorrectly classified by j (case 1), $P(i = 1|j = 1)$ as the number of instances correctly classified by i over the number of instances correctly classified by j (case 2). Analogously, $P(i = 0|j = 1)$ can be considered the number of instances misclassified by i over the number of instances correctly classified by j (case 3) and $P(i = 0|j = 0)$ as the number of instances misclassified by i over the number of instances misclassified also by j (case 4).

The contribution r_i^C of each classifier $i \in C$ can be estimated as:

$$r_i^C = \frac{\sum\limits_{j \in \{C \setminus i\}} \sum\limits_{k \in \{0,1\}} P(i = 1|j = k)P(j = k)}{\sum\limits_{j \in \{C \setminus i\}} \sum\limits_{k \in \{0,1\}} P(i = 0|j = k)P(j = k)} \tag{4}$$

where $P(j = k)$ is the prior of classifier j to either correctly or incorrectly predict labels. In particular, $P(j = 1)$ denotes the percentage of correctly classified instances (i.e. accuracy), while $P(j = 0)$ represents the rate of misclassified (i.e. error rate). Note that r_i^C depends on the ensemble C of classifiers: starting from

an initial set C, r_i^C is iteratively computed excluding at each iteration the classifier that achieves the lowest r_i^C. In order to define the initial ensemble, the baseline classifiers in C have to show some level of dissimilarity. This can be achieved using models that belong to different families (i.e. generative, discriminative and large-margin models). The proposed strategy allows us to reduce the search space from $\sum_{k=1}^{n} \frac{n!}{k!(n-k)!}$ to $n-1$ potential candidates for determining the optimal ensemble. In fact, at each iteration the classifier with the lowest r_i^C is disregarded until the smallest combination is achieved.

The baseline classifiers considered in this paper are the following:

Dictionary-Based. A Dictionary-based classifier is the simplest and naive method for the polarity classification task. Given two dictionaries, one for negative and one for positive terms, the sentence polarity is determined, checking if each sentence term belongs to the positive or the negative dictionary and finally using the following aggregation function:

$$l(s) = \begin{cases} \text{positive} & \text{if } \#\text{tokens}_+ > \#\text{tokens}_- \\ \text{negative} & otherwise \end{cases} \tag{5}$$

Naïve Bayes. NB [7] is the simplest generative model that can be applied to the polarity classification task. It predicts the polarity label l given a vector representation of textual cues by exploiting the Bayes' Theorem.

Maximum Entropy. ME [8] is a discriminative model that has been largely adopted in the state of the art for polarity classification. It makes no assumption about the relationships between textual cues, which are modeled through several feature functions that can eventually be overlapping and non-independent. In this study, the ME model is trained with feature functions that represent the unigrams within a sentence.

Support Vector Machines. SVMs [9] are linear learning machines that try to find the optimal hyperplane discriminating samples of different classes, ensuring the widest margin.

Conditional Random Fields. CRFs [10] are a type of discriminative probabilistic graphical model. In this work, a linear-chain CRF has been applied at sentence level in order to model the sentiment flow within a paragraph when it is seen as a sequence of dependent sentences. Each sentence, which is assumed to be composed of a sequence of unigrams, is evaluated according to a set of binary feature functions able to capture local properties.

4 Experimental Investigation

4.1 Experimental Setup

In this study, three benchmark datasets are considered.

The first one is "Finegrained Sentiment Dataset, Release 1"[1] (ProductData) [11], and relates to product reviews about books, dvds, electronics, music and video games. Although the original dataset is composed of 5 levels of polarity (POS, NEG, NEU, MIX and NR), a reduction of instances has been performed in order to deal only with positive and negative opinions. The resulting dataset is unbalanced, composed of 1320 ($\simeq 58.84\%$) negative and 923 ($\simeq 41.16\%$) positive reviews.

The second dataset is "Multi-Domain Sentiment Dataset"[2] (ProductDataMD) [12] and contains product reviews taken from Amazon.com about many product types (domains). Reviews contain star ratings (1 to 5 stars) that have been converted into nominal labels ('neg' for ratings lower than 3, 'neu' for ratings equal to 3 and 'pos' for ratings greater than 3). In this study, reviews from category 'Music' and 'Books' are studied separately. ProductDataMD is balanced, composed of 2000 reviews for each of the two categories.

The third dataset, known as "Sentence polarity dataset v1.0"[3] (MovieData) [13], is composed of 10662 snippets of movie reviews extracted from Rotten Tomatoes[4]. The main characteristics of this dataset, which comprises only positive and negative sentences, are related to the informal language adopted (slang and short forms) and to the presence of noisy polarity labeling.

A 10-folds cross validation has been adopted as evaluation criteria.

The setup of the experimental phase relates to classifier settings. The dictionary-based classifier exploits the polarity dictionary originally created by Hu and Liu[5] (DictHuLiu) [14]. This lexicon is composed of 4783 negative and 2006 positive words. Terms that are not included neither in the positive nor in the negative dictionary are not considered during the aggregation process. DictHuLiu was defined starting from a small seed set of opinion words, concerning the domain of product reviews, then expanded it exploiting iteratively WordNet's synsets and relationships to acquire other opinion words (also morphological variants and slang words).

For NB and SVM, although different sentence representations (boolean, tf, tf-idf) have been considered, we report only the results based on the "best" weighting schema. To this purpose, only the investigations based on the boolean representation are shown. For training SVM, a linear kernel has been assumed while the training of NB assumes a multinomial distribution. NB and SVM experiments are based on *LingPipe*[6] and *LIBSVM*[7] respectively. Concerning ME and CRF, the models have been trained by maximizing the likelihood until convergence. In particular, ME is trained using multiple conditional likelihood

---

[8] and CRF is induced exploiting regularized likelihood [10]. ME and linear chain CRF classifiers have been applied using the MALLET package[8].

4.2   Computational Results

In this section the performance achieved on the considered datasets, both by the baseline classifiers and the ensemble methods (MV and BMA, described in Sect. 2), are presented. To this purpose, we measured Precision (P), Recall (R) and F_1-measure, defined as

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad R = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{6}$$

both for the positive and negative labels (in the sequel denoted by P_+, R_+, $F1_+$ and P_-, R_-, $F1_-$ respectively). We also measured Accuracy, defined as

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{7}$$

Table 1. Performance of the baseline classifiers on ProductData

ProductData				
	DIC	NB	ME	SVM	CRF
P_-	**0.7561**	0.7024	0.7033	0.6880	0.7292
R_-	0.7545	0.7250	0.7758	**0.8326**	0.7621
$F1_-$	**0.7548**	0.7121	0.7360	0.7530	0.7443
P_+	0.6469	0.5872	0.6203	**0.6550**	0.6375
R_+	**0.6478**	0.5565	0.5239	0.4565	0.5924
$F1_+$	**0.6463**	0.5689	0.5638	0.5364	0.6120
Acc	**0.7107**	0.6558	0.6723	0.6781	0.6924

Table 2. Performance of DIC (best classifier), MV and BMA for the ensemble {DIC, ME, CRF} on ProductData

ProductData		
	DIC	MV	BMA
P_-	0.7561	0.7603	**0.7703**
R_-	0.7545	0.8265	**0.8447**
$F1_-$	0.7548	0.7912	**0.8050**
P_+	0.6469	0.7136	**0.7401**
R_+	**0.6478**	0.6217	0.6348
$F1_+$	0.6463	0.6621	**0.6813**
Acc	0.7107	0.7424	**0.7585**

Table 1 reports performance achieved on ProductData. As expected, performance on negative cases are higher than performance achieved on positive cases, because ProductData is unbalanced (1320 negative and 923 positive instances). This is also confirmed by MV and BMA, which achieve high recall on the negative cases and low recall on the positive ones (Table 2). The best classifier DIC

Table 3. Computation of r_i^C and accuracy ensembles on ProductData

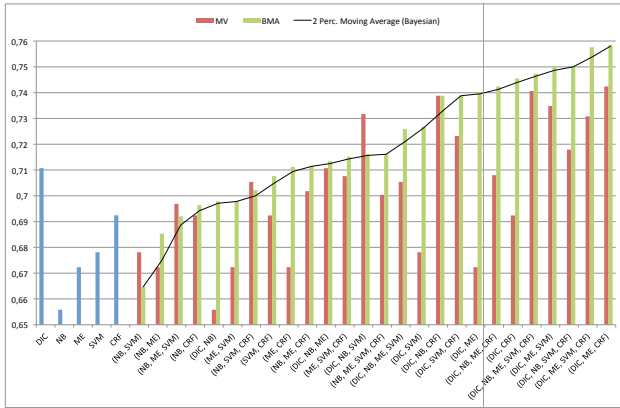| Iteration	DIC	NB	ME	SVM	CRF	Accuracy
1	2.154	**1.572**	1.648	1.641	1.783	0.747
2	2.111	-	1.678	**1.593**	1.735	0.757
3	2.131	-	**1.676**	-	1.870	**0.758**
4	2.123	-	-	-	**1.918**	0.745

---

[8] http://mallet.cs.umass.edu/

Fig. 1. Accuracy of baseline classifiers, MV and BMA on ProductData

achieves 71.07% of global accuracy (Table 1), while the best ensemble (composed of DIC, ME and CRF) achieves an accuracy of 74.24% and 75.85% for MV and BMA respectively. The contribution of each classifier belonging to a given ensemble can be computed a priori by applying the model selection strategy.

Starting from the initial set C={DIC, NB, ME, SVM, CRF}, the classifiers are sorted with respect to their contribution by computing (4). As shown in Table 3, the classifier with the lowest contribution at the first iteration is NB. Then, (4) is re-computed on the ensemble {C \ NB}, highlighting SVM as the classifier with the lowest contribution. At iteration 3 and 4, the worst classifiers to be removed from the ensemble are ME and CRF respectively.

As highlighted by the accuracy measure, the model selection heuristic is able to determine the optimal composition by evaluating four ensemble candidates. In this case, the optimal solution is found at iteration 3, where the best ensemble is
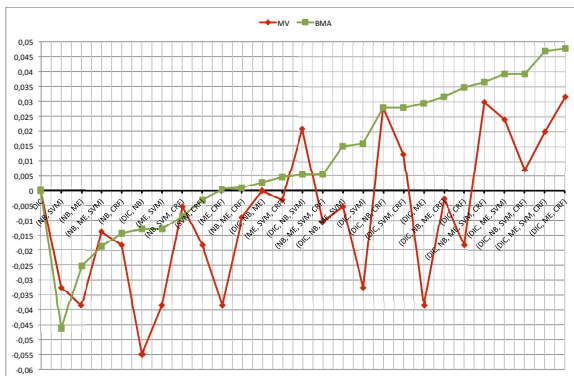

Fig. 2. Cumulative chart of accuracy on ProductData

Table 4. Performance of the baseline classifiers on ProductDataMD "books"

Table 5. Performance of the baseline classifiers on ProductDataMD "music"

ProductDataMD on Books					
	DIC	NB	ME	SVM	CRF
P_-	0.7381	0.6564	0.7387	0.7470	**0.8041**
R_-	0.5640	0.7060	**0.8100**	0.7560	0.7550
$F1_-$	0.6383	0.6795	0.7713	0.7502	**0.7781**
P_+	0.6471	0.6837	**0.7943**	0.7538	0.7696
R_+	0.7980	0.6300	0.7130	0.7410	**0.8150**
$F1_+$	0.7142	0.6547	0.7495	0.7458	**0.7912**
Acc	0.6810	0.6680	0.7615	0.7485	**0.7850**

ProductDataMD on Music					
	DIC	NB	ME	SVM	CRF
P_-	0.8042	0.6652	**0.8107**	0.7251	0.7894
R_-	0.4610	0.6360	0.7010	0.7230	**0.7550**
$F1_-$	0.5839	0.6495	0.7508	0.7233	**0.7711**
P_+	0.6229	0.6526	0.7383	0.7249	**0.7663**
R_+	**0.8870**	0.6800	0.8360	0.7250	0.7980
$F1_+$	0.7314	0.6654	**0.7834**	0.7241	0.7813
Acc	0.6740	0.6580	0.7685	0.7240	**0.7765**

composed of {DIC, ME, CRF}. For sake of completeness, all ensemble performance are depicted in Figure 1. The cumulative chart of accuracy is reported in Figure 2.

Table 4 reports performance achieved on ProductDataMD "books". The contribution of the best BMA is about 3.55%, while 1.6% for MV.

Fig. 3. Accuracy of baseline classifiers, MV and BMA on ProductDataMD "books"
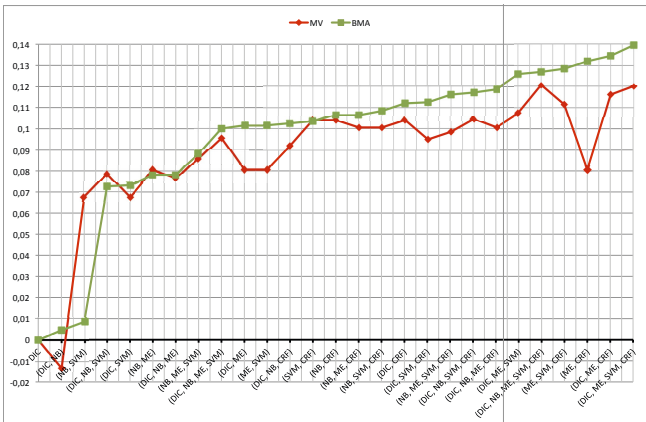

Fig. 4. Cumulative chart of accuracy on ProductDataMD "books"

As shown in Table 6, also in this case the optimal ensemble is determined within the search space of four ensemble candidates. According to the proposed heuristic, the optimal combination is composed of {DIC, ME, SVM, CRF} found at iteration 2. This result can be also validated looking at Figure 3 and 4.

Table 6. Computation of r_i^C and accuracy ensembles on ProductDataMD "books"

Iteration	DIC	NB	ME	SVM	CRF	Accuracy
1	1.859	**1.654**	2.473	2.117	2.644	0.808
2	**1.816**	-	2.480	1.959	2.460	**0.820**
3	-	-	2.315	**1.696**	2.165	0.809
4	-	-	**2.219**	-	2.616	0.813

Performance achieved on ProductDataMD "music" are reported in Table 5. The classifier which achieves the highest performance is CRF with 77.65%. Concerning the voting paradigms, while the accuracy of the best MV is 79.35%, BMA is able to guarantee performance of 80.3%.
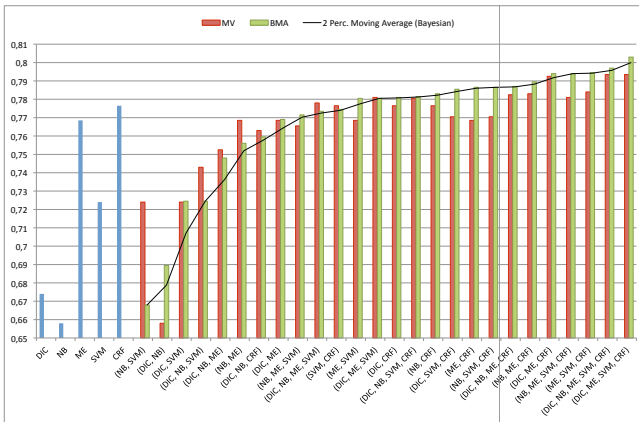

Fig. 5. Accuracy of baseline classifiers, MV and BMA on ProductDataMD "music"

This result can be easily figured out by Table 7, where the best ensemble for BMA is composed of DIC, ME, SVM and CRF (Iteration 2).

Table 7. Computation of r_i^C and accuracy ensembles on ProductDataMD "music"

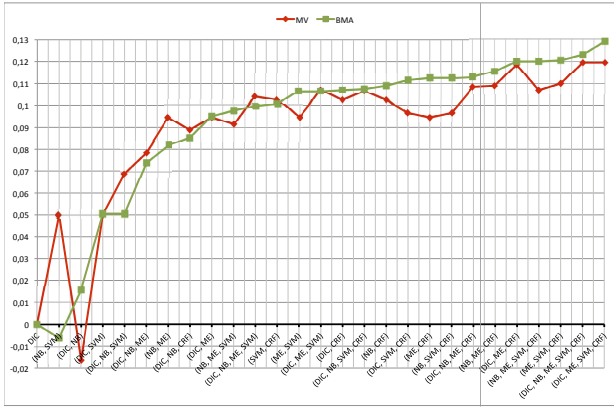| Iteration	DIC	NB	ME	SVM	CRF	Accuracy
1	1.700	**1.579**	2.565	1.982	2.608	0.797
2	**1.638**	-	2.558	1.852	2.457	**0.803**
3	-	-	2.537	**1.644**	2.211	0.794
4	-	-	**2.337**	-	2.472	0.786

Fig. 6. Cumulative chart of accuracy on ProductDataMD "music"

Table 8 shows performance achieved by the baseline classifiers on MovieData. As mentioned in Sect. 4.1, the opinion words belonging to the dictionary are concerned with the product reviews. This explains why DIC obtains low performance on this dataset. Although DIC is the classifier with the worst performance, the outperforming ensemble is composed of DIC, ME and CRF (78.55% of accuracy by BMA and 78.08% by MV). This highlights again that the best ensemble is not necessarily composed of the classifiers which individually lead to the highest performance.

Table 8. Performance of the baseline classifiers on MovieData

MovieData					
	DIC	NB	ME	SVM	CRF
P_-	0.6313	0.7104	0.7638	0.7395	**0.7711**
R_-	0.7325	0.7023	**0.7816**	0.7559	0.7409
$F1_-$	0.6780	0.7062	**0.7713**	0.7475	0.7555
P_+	0.6810	0.7057	**0.7738**	0.7504	0.7508
R_+	0.5717	0.7135	0.7523	0.7334	**0.7795**
$F1_+$	0.6214	0.7095	0.7609	0.7417	**0.7647**
Acc	0.6521	0.7079	**0.7670**	0.7447	0.7602

In fact, the improvement with respect to the best classifier ME (76.70%) is close to 2% for MV and 3.55% for BMA (Figure 7 and 8). The selection of the best BMA ensemble can be easily derived from the application of the proposed heuristic reported in Table 9.

In conclusion, Figure 9 shows that BMA, together with the model selection strategy, ensures a significant performance improvement with regard to the studied baseline classifiers and MV.

5   Conclusion

In this work we discussed how to explore the potential of ensembles of classifiers for sentence level polarity classification and proposed an ensemble method
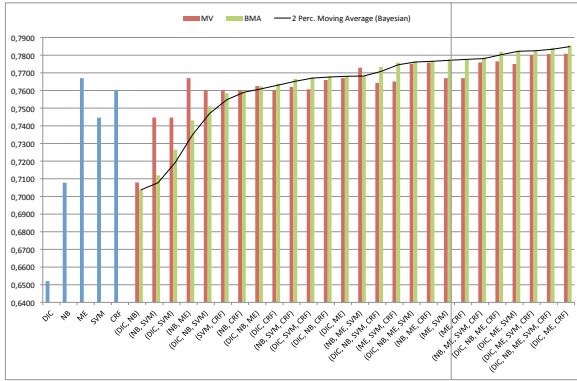
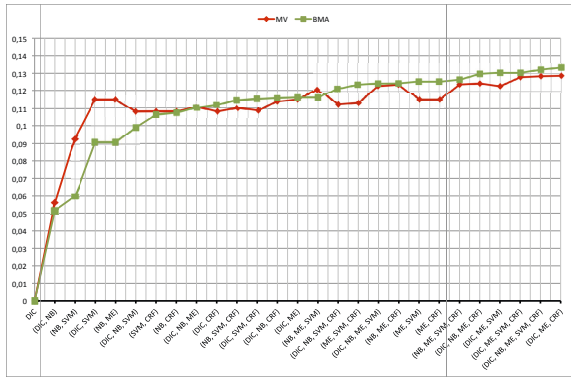**Fig. 7.** Accuracy of baseline classifiers, MV and BMA on MovieData

Fig. 8. Cumulative chart of accuracy improvement on MovieData

Table 9. Computation of r_i^C and accuracy ensembles on MovieData

Iteration	DIC	NB	ME	SVM	CRF	Accuracy
1	1.619	**1.602**	2.121	1.706	1.948	0.784
2	1.584	-	2.132	**1.580**	1.847	0.782
3	**1.586**	-	2.251	-	2.147	**0.785**
4	-	-	1.742	-	**1.648**	0.777

based on Bayesian Model Averaging. We further proposed an heuristic aimed at evaluating the a priori contribution of the single models to the classification task that can be used to help in selecting the best ensemble composition. The experimental results show that the proposed solution is particularly effective and efficient, thanks to the ability to a priori define a strategic combination of different classifiers. An ongoing research is the extension of BMA to a wider range of labels. We are considering a hierarchical voting framework where the discrimination between 'objective' and 'subjective' is firstly addressed, to then approach the polarity classification of subjective expressions.
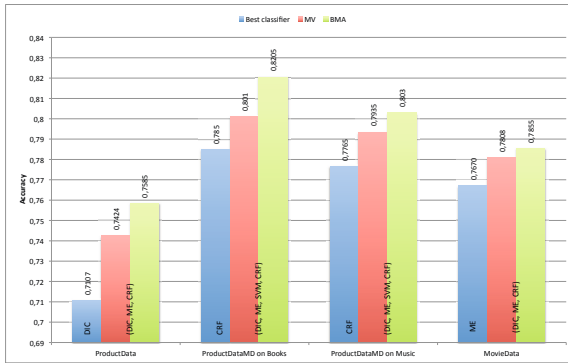
Fig. 9. Summary of accuracy comparison

References

1. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2, 1–135 (2008)
2. Yessenalina, A., Yue, Y., Cardie, C.: Multi-level structured models for document-level sentiment classification. In: Proc. of the Conf. on Empirical Methods in NLP (2010)
3. Dietterich, T.G.: Ensemble learning. In: The Handbook of Brain Theory and Neural Networks, pp. 405–508. Mit Pr. (2002)
4. Whitehead, M., Yaeger, L.: Sentiment mining using ensemble classification models. In: Sobh, T. (ed.) Innovations and Advances in Computer Sciences and Engineering, pp. 509–514. Springer Netherlands (2010)
5. Xiao, M., Guo, Y.: Multi-view adaboost for multilingual subjectivity analysis. In: 24th Inter. Conf. on Computational Linguistics, COLING 2012, pp. 2851–2866 (2012)
6. Hoeting, J.A., Madigan, D., Raftery, A.E., Volinsky, C.T.: Bayesian model averaging: A tutorial. Statistical Science 14(4), 382–417 (1999)
7. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI 1998 Workshop on Learning for Text Categ., pp. 41–48 (1998)
8. McCallum, A., Pal, C., Druck, G., Wang, X.: Multi-conditional learning: Generative/discriminative training for clustering and classification. In: AAAI, pp. 433–439 (2006)
9. Cortes, C., Vapnik, V.: Support-vector networks. ML 20(3), 273–297 (1995)
10. Sutton, C.A., McCallum, A.: An introduction to conditional random fields. Foundations and Trends in ML 4(4), 267–373 (2012)
11. Täckström, O., McDonald, R.: Semi-supervised latent variable models for sentence-level sentiment analysis. In: Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 569–574 (2011)
12. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Association for Computational Linguistics (2007)
13. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proc. of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 115–124 (2005)
14. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proc. of the 10th ACM SIGKDD Inter. Conf. on Knowledge Discovery and DM, pp. 168–177 (2004)

An Approach for Extracting and Disambiguating Arabic Persons' Names Using Clustered Dictionaries and Scored Patterns

Omnia Zayed, Samhaa El-Beltagy, and Osama Haggag

Center of Informatics Science, Nile University, Giza, Egypt
{omnia.zayed,samhaaelbeltagy,osama.haggag}@gmail.com

Abstract. Building a system to extract Arabic named entities is a complex task due to the ambiguity and structure of Arabic text. Previous approaches that have tackled the problem of Arabic named entity recognition relied heavily on Arabic parsers and taggers combined with a huge set of gazetteers and sometimes large training sets to solve the ambiguity problem. But while these approaches are applicable to modern standard Arabic (MSA) text, they cannot handle colloquial Arabic. With the rapid increase in online social media usage by Arabic speakers, it is important to build an Arabic named entity recognition system that deals with both colloquial Arabic and MSA text. This paper introduces an approach for extracting Arabic persons' name without utilizing any Arabic parsers or taggers. Evaluation of the presented approach shows that it achieves high precision and an acceptable level of recall on a benchmark dataset.

1 Introduction

Named entity recognition (NER) has become a crucial constituent of many natural language processing (NLP) and text mining applications. Examples of those applications include Machine Translation, Text Clustering and Summarization, Information Retrieval and Question Answering systems. An exhaustive list can be found in [5]. Arabic NER has attracted much attention during the past couple of years, with research in the area achieving results comparable to those reported for the English language.

Approaches for recognizing named entities from text have been divided into three categories which are "Rule Based NER", "Machine learning based NER" and "Hybrid NER". The "Rule Based NER" combines grammar, in the form of handcrafted rules, with gazetteers to extract named entities. "Machine learning based NER" utilizes large datasets and features extracted from text, to train a classifier in order to recognize a named entity. Hence this approach converts the named recognition task into a classification task. Machine learning algorithms could be further divided into either supervised or unsupervised. The "Hybrid NER" combines the machine learning approach with the rule based approach. A comparison between the rule based approach and the machine learning approach is given in [13]. As mentioned in [1, 13, 17], it is difficult to extend the rule based approach to new domains because of the necessity of

complicated linguistic analysis to detect the named entities. Conversely, the difficulty of the machine learning approach lies in that it requires a precise selection of features from a training dataset which is tagged in a certain manner to recognize new entities from new testing dataset in the same domain.

To reach acceptable results however, employment of an Arabic parser is a must in any of the above listed approaches. While this is perfectly valid for extracting named entities from MSA, it is difficult to apply on colloquial Arabic, which is currently used extensively in micro-blogging and social media contexts. The main difficulty of applying previously devised approaches on this type of media, is the fact that existing Arabic parsers cannot deal with colloquial Arabic at any acceptable degree of accuracy. Without the utilization of such parsers, the degree of ambiguity in Arabic person name detection rises significantly for reasons that are detailed in section 2.

This paper introduces an approach for extracting Arabic persons' names, the most challenging Arabic named entity, without utilizing any Arabic parsers or taggers. The presented approach makes use of a limited set of dictionaries integrated with a statistical model based on association rules, a name clustering module, and a set of rules to detect person names. The main challenges addressed by this work could be summarized as:

— Overcoming the person name ambiguity problem without the use of parsers, taggers or morphological analyzers.
— Avoiding the shortcomings of both rule based NER and machine learning based NER approaches including employment of complex linguistic analysis, huge sets of gazetteers, huge training sets, feature extraction from annotated corpus…etc. in order to be able to extend the approach to new domains, primarily colloquial Arabic, in our future work.

Evaluation of the presented approach was carried out on a benchmark dataset and shows that the system outperforms the state of the art machine learning based system. While the recall of the system falls below the state of the art hybrid system, the precision of the system is comparable to it.

The rest of the paper is organized as follows: Section 2 discusses Arabic specific challenges faced when building NER systems; Section 3 describes the proposed approach in detail. In Section 4, system evaluation on a benchmark dataset is discussed. Section 5 highlights an overview of the literature on NER systems in Arabic language. Finally conclusion and future work is presented in Section 6.

2   Arabic Specific Challenges for Persons' Names   Recognition

The Arabic language is a complex and rich language, which steps up the challenges faced by researchers when developing an Arabic natural language processing (ANLP) application [11]. Recognizing Arabic named entities is a difficult task due to a variety of reasons as explained in detail in [1, 11]. Those reasons are revisited with examples:

— One of the major challenges of Arabic language is that it has many levels of ambiguity [11]. A significant level of ambiguity is the semantic ambiguity in which one word could imply a variety of meanings. For example, the word "نبيه" could imply the phrase (his prophet), the adjective (intelligent) or the name of a person (Nabih).

— Arabic named entities could appear with conjunctions or other connection letters which complicates the task of extracting persons' names from Arabic text such as "ومحمد" (and Mohammed), "كمحمد" (as Mohammed), "لمحمد" (to Mohammed), "فمحمد" (then Mohammed) or "بمحمد" (with Mohammed).

— Most of the Arabic text suffers from lack of diacritization. Lack of diacritization causes another level of ambiguity in which a word could belong to more than one part of speech with different meanings [1, 11]. For example, the word "نهي" without diacritics could imply the female name (Noha), or the verb (prohibited).

— Arabic lacks capitalization as it has a unified orthographic case [1]. In English some named entities can be distinguished because they are capitalized. These include persons' names, locations and organizations.

— Arabic text often contains not only Arabic named entities, but translated and transliterated named entities to Arabic [11] which often lack uniform representation. For example, the name (Margaret) can be written in Arabic in different ways such as "مرجريت","مارجريت", "مرغريت" or "مارغريت".

— Many persons' names are either derived from adjectives or can be confused with other nouns sharing the same script. Examples of ambiguous Arabic male names include [Adel, Said, Hakim, and Khaled] their different adjective or noun polysemy are [Just, Happy, Wise, and Immortal]. Examples of some ambiguous female names include [Faiza, Wafia, Omneya, and Bassma] which could be interpreted as [Winner, Loyal, Wish, and Smile]. Examples of some ambiguous family/last names are [Harb, Salama, Khatab and Al-Shaer] which translate to [War, Safety, Speech/Letter and The Poet].

— Moreover, some Arabic persons' names match with verbs such as [Yahya, Yasser, and Waked] their different verb polysemy are [Greets, Imprisons, and Emphasized]. In addition, some foreign persons' names transliterated to Arabic could be interpreted as prepositions or pronouns such as [Ho, Anna, Ann, and, Lee] their different prepositions or pronouns are [He, I, That, Mine].

The combination of the above listed factors, makes the recognition of Arabic person names the most challenging of Arabic named entities to extract without any parsers. Simply building a system based on straightforward matching of persons' names using dictionaries, will often result in mistakes. The traditional solution for this is using parsers or taggers. However, extracting persons' names from colloquial Arabic text invalidates this solution as existing parsers fail to parse colloquial Arabic at an acceptable level of precision mainly due to sentence irregularity, incompleteness and the varied word order of colloquial Arabic [17]. In this paper, the ambiguity problem is addressed in two ways. First, publicly available dictionaries of persons' names are grouped into clusters. Second, a statistical model based on association rules is built to extract patterns that indicate the occurrence of persons' names. These approaches will be explained in detail in section 3.

3   The Proposed Approach

In this work, a rule based approach combined with a statistical model, is adopted to identify and extract person names from Arabic text. Our approach tries to overcome two of the major shortcomings of using rule based techniques which are the difficulty of modifying a rule based approach for new domains and the necessity of using huge sets of gazetteers. Section 5 highlights the differences between the resources needed by our approach and previous approaches.

Our approach consists of two phases, as shown in Fig. 1. In the first phase, "The building of resources phase", person names are collected and clustered, and name indicating patterns are extracted.   In the second phase, "Extraction of persons' names phase", name patterns and clusters are used to extract persons' names from input text. Both of these phases are described in depth, in the following subsections.
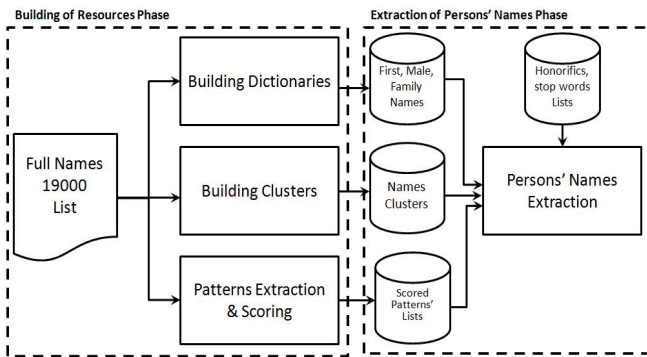

Fig. 1. System Architecture

3.1   The Building of Resources Phase

In this phase the resources on which the system depends are prepared. This phase is divided into 4 stages. In the first stage, persons' names are collected from public resources. In the second stage, dictionaries of first, male/middle and family persons' names are built from collected resources. In the third stage, names are grouped together into clusters to address the Arabic persons' names ambiguity problem as will be detailed later.   In the fourth and final stage, a corpus is used to build and score patterns which indicate the occurrence of a person's name. Scoring of the patterns is done using association rules.

Name Collection. Wikipedia[1], with its huge collection of names under the people category, offers an excellent resource for building a database for persons' names. Kooora[2], which is an Arabic website for sports, also provides a large list of football

---

[1] http://ar.wikipedia.org/wiki/تصنيف:تراجم
[2] http://www.kooora.com/default.aspx?showplayers=true

and tennis players' names. In this stage, Wikipedia and Kooora websites were used to collect a list of about 19,000 persons' full names. Since the aim of this work is not just to recognize names of famous people, but instead to identify the name of any person even if it does not appear in the collected lists, the collection was further processed and refined in order to achieve this goal in the "Building the dictionaries" stage.

Building of Dictionaries. In this stage, the list of names collected in the previous stage (we call this list the "full_names_19000_list") was processed in such a way so as to separate first names from family names in order to create three names lists which are first, male/middle, and family names lists. Collecting a list of male names is important as a male name is often used as a family name. It is difficult to know whether a first name is a male or female name, but any middle name is always a male name. At the beginning, input names in the list are normalized using the rules presented in [12]. This step addresses the different variations of Arabic persons' name representation. As described in [17], Arabic names could have affixes such as prefixes or embedded nouns. A word preceded or followed by those affixes must not be split on white spaces, instead the word and its affix should be considered as a single entity. For example, the male name عبد العزيز (Abdulaziz) should not be split as عبد (Abd) denoting the first name and العزيز (Alaziz) denoting a family name, instead it should be treated as single entity عبد العزيز (Abdulaziz) and considered as a first name. Table 1 lists the different variations of Arabic persons' names with examples [17].

Table 1. Different variations of writing Arabic persons' names

Case	Example	Extracted Complex Entity	
Simple case (no affixes)	احمد محمود Ahmad Mahmoud	Not applicable	
Prefix case {عبدAbd, ابوAbou, بن Bin, ال Al, …etc}	عبد العزيز ال سعود Abdulaziz Al Saud	"عبد العزيز"	First Name
		"ال سعود"	Family Name
Double prefix case {بن عبد Bin AbouAbd, ابوعبد Abd, … etc}	سلطان بن عبد العزيز ال سعود Sultan bin Abdulaziz Al Saud	"بن عبد العزيز"	Middle Name
		"ال سعود"	Family Name
Embedded noun case {الدين El-Deen, الله Allah, … etc}	هيردي نور الدين Herdi Noor Al-Din	"نور الدين"	Family Name
Complex name (prefix + embedded noun)	تقي الدين محمد بن معروف Taqi al-Din Muhammad ibnMa'ruf	"تقي الدين"	First Name
		"بن معروف"	Middle Name

Any honorifics or titles preceding or following a name, were removed using a compiled list of honorifics[3] that can precede or follow a name.

Building of Name Clusters. In a simplistic world, once the name lists are built, they can be used to identify previously unseen names by stating that a full name is composed of a first name followed by zero or more male names followed by (a male name or a family name). However, as stated before, the inherent ambiguity of Arabic names, does not lend itself to such a simplistic solution. One of the problems of simple matching is the possibility of incorrectly extracting a name which is a combination of an Arabic name and a foreign name. For example, given the   phrase: اتهم ايمن بوش (Ayman accused Bush), using a simple matching approach would result in the extraction of the full name ايمن بوش (Ayman Bush) even though it is highly unlikely that an Arabic person's name such as ايمن (Ayman) will appear besides an American person's name such as بوش (Bush).   In the example above, the translation put the verb "accused" between "Ayman" and "Bush", but in the Arabic representation, both names are placed next to each other and preceded by the verb.   Since Arabic text often contains not only Arabic names, but names from almost any country transliterated to Arabic, incorrectly identifying those could affect the system's precision significantly. A more common form of error resulting from simple matching is encountered when prepositions or pronouns match with names in the compiled name lists as explained in section 2. For example when the phrase ان محمد (That Mohammed) is encountered, the simple matching approach will result in the incorrect extraction of the full name: ان محمد (Ann Mohammed).

Given the fact the "full_names_19000_list" contains Arabic, English, French, Spanish, Hindi, and Asian persons' names, written in the Arabic language, we decided to cluster these names and allow name combinations only within generated clusters.

As a pre-processing step, the 19,000 persons' names list is traversed to build a dictionary in which the first name is a key item whose corresponding value is a list of the other middle and family names that have occurred with it. The variations of writing Arabic persons' names mentioned in the previous subsection are considered. This dictionary is converted to a graph, such that first names, middle names and family names form separate nodes. Edges are then established between each first name and its corresponding middle and family names. The resulting graph consisted of 17393 nodes, and 22518 undirected edges.

The Louvain method [9] was then applied to the graph for finding communities within the network. A community in this context is a cluster of names that are related. The Louvain method defines a resolution parameter; this parameter manages the size of communities. The standard resolution parameter p value is 1.0. A smaller value for p results in the generation of smaller communities while a larger value for p results in larger communities. By trying several values for this resolution parameter on   the ANERcorp[4] [3] dataset,   the value of p=7 was found to produce the best results.

The outcome was a set of 1995 clusters. Each name is assigned a class number denoting which community (cluster) it belongs to.

Fig. 2 shows a snapshot of the resulting clusters. It can be observed from visualizing the data that most of the culturally similar names were grouped together; it can be noted that most of the names common in the Arabic-speaking regions were grouped
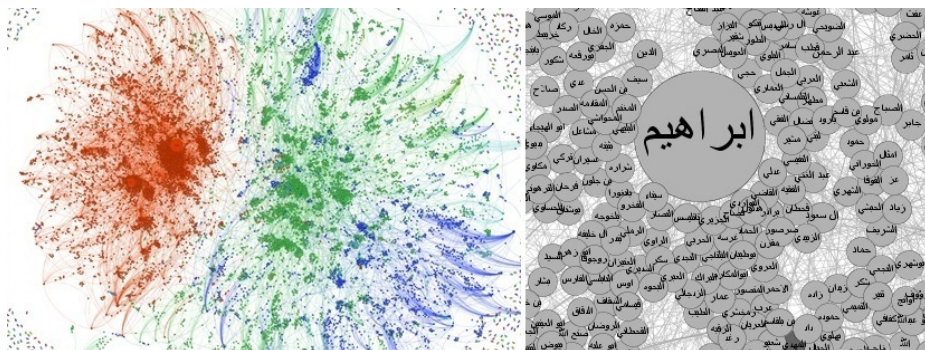
---

[4] `http://www1.ccls.columbia.edu/~ybenajiba/downloads.html`

Fig. 2. Visualization of generated clusters, to the left are all generated clusters, lone clusters can be seen on the border and the two largest clusters are those of Arabic names (left) and Western names (right). To the right is a closer view of a subset of the Arabic names cluster.

together.   The same applies to English and French names and to other names that are kind of unique to their region such as Asian names.

Extracting Scored Patterns. In this stage, a statistical model is built to automatically learn patterns which indicate the occurrence of a person's name.

Initially each name in the "full_names_19000_list" is used as a query to search news articles to build learning dataset from the same domain that we are targeting to extract persons' names from. Akhbarak[5] API and Google Custom Search API[6] were used to search and retrieve news stories.

Around 200 news article links were crawled (whenever possible) for each person name in the "full_names_19000_list". A total number of around 3,800,000 million links were collected using this procedure. After downloading the pages associated with these links, BoilerPipe[7] was used to extract the content or body of each news article.   Very similar stories were detected and removed.

Following this step, unigram patterns around each name are extracted. Three lists are formed. A complete pattern list keeps set of complete patterns around the name with their count. A complete pattern consists of <word$_1$><name><word$_2$>. The <name> part just indicates that a name has occurred between words: word$_1$ and word$_2$. Two type of unigram pattern lists are kept: a "before" list keeps the patterns that appear before a name with their counts (example: اكد (confirmed)) and an "after" list stores patterns that occur after a name with their count (example: ان (that)).

Finally the support measure employed by association rules [2] is used to score each pattern in the three lists. Support is calculated as the ratio of the count of a pattern followed by a name over the total count of all patterns followed by a name.

The newly created three lists of scored patterns are saved descendingly according to the value of the score.

3.2   Extraction of Persons' Names Phase

The persons' names extraction process is dependent on the previous pre-prepared resources which are the dictionaries of first, middle, and family names, divided into clusters, a list of honorifics, a list of stop words and the patterns lists. Rules are implemented to extract persons' names from the unseen dataset of the same targeted domain. The benchmark dataset, ANERcorp [3] is used to evaluate the proposed system. The system assumes that any full name consists of a first name followed by one or more male names followed by zero or one family name. A family name appearing on its own (Bush for example), must have previously appeared as part of full name within the same text, in order to be extracted. In some text pieces, a part of a full name may appear on its own as in the phrases: واضاف كلينتون (and Clinton added), or قال محمد (Mohammed said In order to be able to disambiguate and extract such names; a list of "disambiguous names" is used. The "disambiguous names" list is a manually created list extracted from our previously created names lists and contains names that do not share the same meanings with other adjectives, nouns …etc.

When extracting names from text, employed rules can be divided into two classes: rules for "learning new names" and rules for "matching known names". In the "matching known names" rules, the generated name clusters are used to ensure that all candidate portions of a name fall in the same cluster to avoid matching mistakes and to solve the ambiguity problems mentioned previously. One of the rules used to "match known names" in the extraction phase is as follows:

```
For each word wᵢin the target text:
  If wᵢin patterns_before_list
    If wᵢ₊₁ in honorific_list
    Check for names from wᵢ₊₂ in the same cluster;
      Stop when a delimiter d is_found where d ∈
  (pattern_after|stop_word|punctuation|title_start)
    Else
  Check for names from wᵢ₊₁ in the same cluster;
  Stop when a delimiter d is_found where d ∈
      (pattern_after|stop_word|punctuation|title_start)
  Else if wᵢ in honorific_list
  Check for names from wᵢ₊₁ in the same cluster;
    Stop when a delimiter d is_found where d ∈
(pattern_after|stop_word|punctuation|title_start)
```

The above rule is used to extract names from a sentence such as:

قال الرئيس محمد مرسيان مصر تخطو ...

President Mohammad Morsi said that Egypt is stepping through …

This rule is generalized to extract names from sentences which contain multi honorifics before the person's name such as:

قال رئيس الوزراء الاسرائيلي ايهود اولمرت إنه عازم ...

Prime Minister of Israel Ehud Olmert said that he will …

An example of one of the rules used to "learn new names" is to check for a pattern from "the patterns before list" followed by an unknown name (not in the dictionaries) with the prefix عبد (Abd) followed by known male name and/or family name (the previous stopping criterion is used).

Another rule to learn new unknown family names is to check for a pattern from "the patterns before list" followed by a known first name followed by an unknown name such as:

وقال *مدير المؤسسه* فريدون موافقان المستثمر …

The Director of the Foundation Feridun Mouafiq said that the investor …

In this example فريدون (Feridun) is a known first name while موافق (Mouafiq) is unknown family name; our system is able to extract this person's full name correctly.

Other rules are employed, but are not included due to space limitations. The next section shows how the use of patterns and the use of clusters improve the system performance.

4   System Evaluation

The presented system was evaluated using the precision, recall and f-score measures based on what it extracted as names from the benchmark ANERcorp [3] dataset. As mentioned in [3] , ANERcorp consists of 316 articles which contain 150,286 tokens and 32,114 types. Proper Names form 11% of the corpus. Table 2 provides a comparison between the results of the presented system with two state of the art systems which are the hybrid NERA approach [1] and the machine learning approach using conditional random fields (CRF) [4].

Table 2. Comparison between our system performance in terms of precision, recall and F-score with the current two state of the art systems

	Precision	Recall	F-score
Hybrid System	94.9	90.78	92.8
CRF System	80.41	67.42	73.35
Our System	**93.22**	**78.88**	**85.45**

From this comparison, it can be inferred that our system outperforms the state of the art machine learning system. However the recall of our system is still below the recall of the state of the art hybrid approach. Our system still needs some improvements to compete with the hybrid NERA approach.

Table 3. Effect of individual system's   components on overall system performance

	Precision	Recall	F-score
Dictionaries Only	71.0	62.98	66.75
Dictionaries+ Clusters	77.24	58.62	66.65
Dictionaries+ Clusters+ Patterns	94.96	76.91	84.99
Dictionaries+ Clusters+ Patterns+ Disambiguation list	**93.22**	**78.88**	**85.45**

Table 3 shows the effect of using clusters, patterns and disambiguation lists on the system's performance.

5   Related Work

The majority of previous work addressing NER in Arabic language was developed for the formal MSA text which is the literary language used in newspapers and scientific books. NER from informal colloquial Arabic, currently being used widely in social media communication, has not been directly addressed. In [17], previous work on Arabic NER is discussed extensively. The currently used rule based approaches to extract named entities from MSA text, are dependent on tokenizers, taggers and parsers combined with a huge set of gazetteers. Although, those approaches might be for extracting persons' names from a formal domain, it will be hard to modify them for the colloquial domain [17].

There is some similarity between our approach and another approach based on local grammar [16] which uses reporting verbs as patterns to indicate the occurrence of persons' names. However our approach extracts patterns automatically from the domain under study, while the other approach is limited to a list of reporting verbs. NERA [15] is a system for extracting Arabic named entities using a rule-based approach in which linguistic grammar-based techniques are employed. NERA was evaluated on purpose-built corpora using ACE and Treebank news corpora that were tagged in a semi-automated way. The work presented in [10] describes a person named entity recognition system for the Arabic language. The system makes use of heuristics to identify person names and is composed of two main parts: the General Architecture for Text Engineering (GATE) environment and the Buckwalter Arabic Morphological Analyzer (BAMA). The system makes use of a huge set of dictionaries.

As mentioned in [1], the most frequently used approach for NER is the machine learning approach by which text features are used to classify the input text depending on an annotated dataset. Benajiba et al. applied different machine learning techniques [3–8] to extract named entities from Arabic text. The best performing of these makes use of optimized feature sets [4]. ANERSys [3] was initially developed based on n-grams and a maximum entropy classifier. A training and test corpora (ANERcorp) and gazetteers (ANERgazet) were developed to train, evaluate and boost the implemented technique. ANERcorp is currently considered the benchmark dataset for testing and evaluating NER systems. ANERSys 2.0 [7] basically improves the initial technique used in ANERSys by combining the maximum entropy with POS tags information. By changing the probabilistic model from Maximum Entropy to Conditional Random Fields the accuracy of ANERSys is enhanced [8].

Hybrid approaches combine machine learning techniques, statistical methods and predefined rules. The most recent hybrid NER system for Arabic uses a rule based NER component integrated with a machine learning classifier [1] to extract three types of named entities which are persons, locations and organizations. The reported results of the system are significantly better than pure rule-based systems and pure machine-learning classifiers. In addition the results are also better than the state of the

art Arabic NER system based on conditional random fields [4]. The system was extended to include more morphological and contextual features [14] and to extract eleven different types of named entities using the same hybrid approach.

Compared with other approaches, our system utilizes a far more limited set of resources. All our system requires is a large set of names, which can be easily obtained from public resources such as Wikipedia and a list of honorifics. Our system also, avoids the use of parsers or taggers and the need for annotated datasets.

6   Conclusion and Future Work

This paper presented a novel approach for extracting persons' names from Arabic text. This approach integrated name dictionaries and name clusters with a statistical model for extracting patterns that indicate the occurrence of persons' names. The used approach overcomes major limitations of the rule based approach which are the need for a huge set of gazetteers and domain dependence. More importantly, the fact that the presented work uses no parsers or taggers, and uses publicly available resources to learn patterns, means that the system can be easily adapted to work on colloquial Arabic or new domains. Our rule based approach was able to overcome the ambiguity of Arabic persons' names using clusters. Building the patterns' statistical model using association rules improved the tasks of Arabic persons' names disambiguation and extraction from any domain. System evaluation on a benchmark dataset, showed that the performance of the presented technique is comparable to the state of the art machine learning approach while it still needs some improvements to compete with the state of the art hybrid approach.

This work is a part of a continuous work to extract named entities from any type of Arabic text whether it is the informal colloquial Arabic or the formal MSA. Our plans for the future are to improve the results obtained by this approach while avoiding model over-fitting. The main intention is to test this approach on a colloquial dataset collected from Arabic social media.

References

1. Abdallah, S., Shaalan, K., Shoaib, M.: Integrating rule-based system with classification for Arabic named entity recognition. In: Gelbukh, A. (ed.) CICLing 2012, Part I. LNCS, vol. 7181, pp. 311–322. Springer, Heidelberg (2012)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, SIGMOD 1993, New York, pp. 207–216 (1993)
3. Benajiba, Y., Rosso, P., BenedíRuiz, J.M.: ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp. 143–153. Springer, Heidelberg (2007)
4. Benajiba, Y., Diab, M., Rosso, P.: Arabic named entity recognition using optimized feature sets. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, pp. 284–293. Association for Computational Linguistics, Morristown (2008)

5. Benajiba, Y., Diab, M., Rosso, P.: Arabic named entity recognition: A feature-driven study. IEEE Transactions on Audio, Speech, and Language Processing 17(5), 926–934 (2009)
6. Benajiba, Y., Diab, M., Rosso, P.: Arabic named entity recognition: An svm-based approach. In: The International Arab Conference on Information Technology, ACIT 2008 (2008)
7. Benajiba, Y., Rosso, P.: Anersys 2.0: Conquering the ner task for the Arabic language by combining the maximum entropy with pos-tag information. In: IICAI, pp. 1814–1823 (2007)
8. Benajiba, Y., Rosso, P.: Arabic named entity recognition using conditional random fields. In: Workshop on HLT & NLP within the Arabic World. Arabic Language and Local Languages Processing: Status Updates and Prospects (2008)
9. Blondel, V.D., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 10008 (2008)
10. Elsebai, A., Meziane, F., Belkredim, F.Z.: A rule based persons names Arabic extraction system. In: The 11th International Business Information Management Association Conference, IBIMA 2009, Cairo, pp. 1205–1211 (2009)
11. Farghaly, A., Shaalan, K.: Arabic natural language processing: Challenges and solutions. ACM Transactions on Asian Language Information Processing 8(4), 1–22 (2009)
12. Larkey, L., Ballesteros, L., Connell, M.E.: Light stemming for Arabic information retrieval. Arabic Computational Morphology 38, 221–243 (2007)
13. Mansouri, A., Affendey, L.S., Mamat, A.: Named entity recognition using a new fuzzy support vector machine. In: Proceedings of the 2008 International Conference on Computer Science and Information Technology, ICCSIT 2008, Singapore, pp. 24–28 (2008)
14. Oudah, M., Shaalan, K.: A pipeline Arabic named entity recognition using a hybrid approach. In: Proceedings of the 24th International Conference on Computational Linguistics, COLING 2012, India, pp. 2159–2176 (2012)
15. Shaalan, K., Raza, H.: NERA: Named entity recognition for Arabic. Journal of the American Society for Information Science and Technology, 1652–1663 (2009)
16. Traboulsi, H.: Arabic named entity extraction: A local grammar-based approach. In: Proceedings of the International Multiconference on Computer Science and Information Technology, vol. 4, pp. 139–143 (2009)
17. Zayed, O., El-Beltagy, S., Haggag, O.: A novel approach for detecting Arabic persons' names using limited resources. In: Complementary Proceedings of 14th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2013, Greece (2013)

ANEAR: Automatic Named Entity Aliasing Resolution

Ayah Zirikly and Mona Diab

Department of Computer Science
The George Washington University
Washington DC, USA
{ayaz,mtdiab}@gwu.edu

Abstract. Identifying the different aliases used by or for an entity is emerging as a significant problem in reliable Information Extraction systems, especially with the proliferation of social media and their ever growing impact on different aspects of modern life such as politics, finance, security, etc. In this paper, we address the novel problem of Named Entity Aliasing Resolution (NEAR). We attempt to solve the NEAR problem in a language-independent setting by extracting the different aliases and variants of person named entities. We generate feature vectors for the named entities by building co-occurrence models that use different weighting schemes. The aliasing resolution process applies unsupervised machine learning techniques over the vector space models in order to produce groups of entities along with their aliases. We test our approach on two languages: Arabic and English. We study the impact of varying the level of morphological preprocessing of the words, as well as the part of speech tags surrounding the person named entities, and the named entities' distribution in the data set. We create novel evaluation data sets for both languages. NEAR yields better overall performance in Arabic than in English for comparable amounts of data, effectively using the POS tag information to improve performance. Our approach achieves an $F_{\beta=1}$ score of 67.85% and 70.03% for raw English and Arabic data sets, respectively.

1 Introduction

Named Entity Aliasing Resolution is the process where the different instances (aliases and variants) of an entity are detected and recognized as being referents to the same person within large collections of data. An example of this problem is shown in Figure 1 where each cluster contains several aliases for the same person (e.g. *Yasser Arafat, Abou Ammar*). The variation in name aliases can manifest as a difference in spelling (e.g. *Qaddafi, Gazzafi, Qadafi, Qazzafy*), difference in the name mention such as *Mohamed Hosni Mubarak*, vs. *Hosni Mubarak*, or by using a completely different alias such as *Abou Mazen* as an alternate for *Mahmoud Abbas*. Restricting this problem to aliases of famous people leads to a relatively easier resolution process since the aliases are typically publicly known. However, with the proliferation of web based data and social media, we note the pervasive use of aliases by ordinary people. Nowadays, the use of aliases and fake names is increasingly spreading among larger groups of people and becoming more popular due to political (terrorism, revolutions), criminal and privacy reasons. Hence, the ability to recognize and identify the different aliases of an
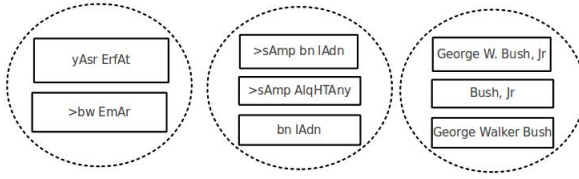
Fig. 1. Personal Named Entities and examples of possible aliases

entity improves the quality of information extracted (higher recall) by helping the entity linking and tracking, leading to better overall information extraction performance.

The NEAR task is relatively close to the Entity Mention Detection (EMD) task.[1,2] However they differ in several aspects. In NEAR there is no processing of pronominal mentions by definition. Moreover, the NEAR task, as defined for this paper, specifically focuses on detecting aliases for person named entities (PNE) and does not handle other NE types such as Organizations and Locations addressed in the EMD task. We should highlight, however, that there is nothing inherent in the NEAR task that bars it from processing other types of NEs. To date, most work in relating PNEs in documents relies on external resources, such as Wikipedia to provide links between aliases and PNE, thus confining the aliasing resolution task to famous people. In this paper, we build a system, Automatic NEAR (ANEAR), that is domain and language independent and does not rely on external knowledge resources. We use unsupervised clustering methods to identify and link the different candidate variants of an entity. We experiment with two languages, Arabic and English, independently. We empirically examine the impact of morphological processing on the feature space. We also investigate the usage of part of speech tag information in our models. Finally, we attempt to measure the effect of various value content modeling approaches on the system such as TF-IDF and co-occurrence frequency. ANEAR's best performance is $F_{\beta=1}$ score is 70.03% on Arabic compared to an $F_{\beta=1}$ score of 67.85% on the English data.

2 Automatic Name Entity Aliasing Resolution (ANEAR) Approach

The underlying assumption for ANEAR is that a person, regardless of his/her number of aliases, can be represented with a finite number of features that identifies him/her. These features encapsulate his/her interests, behaviors, writing style, background, spatial and temporal activities, etc.

The ANEAR system takes as input the unstructured text and generates a feature vector for every PNE as recognized in our data by a Named Entity Recognition (NER) system, i.e. this feature space models the profile for each PNE. The collection of feature vectors produces a Name Features Relatedness (NFR) matrix representing the vector space

---

[1] http://www.itl.nist.gov/iad/mig//tests/ace/ace07/doc/
ace07-evalplan.v1.3a.pdf
[2] http://projects.ldc.upenn.edu/ace/docs/English-Entities-
Guidelines_v6.6.pdf

model. We populate the NFR matrix with different values based on variable weighting schemes that reflect the relatedness scores. Subsequently, we apply unsupervised clustering algorithms to extract and group the different aliases and variants of an entity in one cluster. We experiment with two languages English and Arabic and use parallel data of the same size in order to compare and contrast performance cross-linguistically.

2.1 Building the Name Features Relatedness(NFR) Matrix

The selection of the features in conjunction with the relatedness scoring scheme has a significant impact on the performance of the clustering algorithm. The structure of the matrix is as follows: the row entries of the matrix are the PNEs, the dimensions are either bag of words (BOW) features or classes derived from them such as POS tags, and the feature values are some form of the co-occurrence statistic between the PNE and the feature instance.

2.1.1 Feature Dimensions. Our basic feature set is a BOW feature. We experiment with several possible tokenization levels for the words in the data collection: (i) LEX Inflected forms known as lexemes e.g. *babies* is a lexeme and contractions such as *isn't* are spelled out as *is not*; (ii) LEM Citation forms known as lemmas[3], *babies* is the lexeme and it would be reduced to the lemma *baby*, likewise the lexeme *is* becomes the lemma *be*. It is worth noting that for Arabic, a characteristic of the writing system is that words are typically rendered without short vowels and other pronunciation markers known as diacritics. For our purposes the LEM for Arabic will be the fully diacritized lemma, and the Lexeme, LEX is not diacritized. In order to identify if diacritization helps our process on the lexeme and the lemma levels, we explore a third word form in Arabic which is the diacritized lexeme DLEX. An example of a diacritized lexeme in Arabic is the DLEM *xaAmiso*,[4] *fifth*, and its undiacritized form is *xAms*.

Creating the vector space model for English and Arabic varies due to the nature of the two languages. Arabic has a much more complex morphological structure than English. Hence, as expected the number of lexeme dimensions for Arabic far exceeds that for English. Moreover, the lexeme to lemma ratio in Arabic is much higher in Arabic compared to English. We note that our Arabic data collection has 71910 diacritized lexemes compared to 67125 undiacritized lexeme and 38537 diacritized lemmas corresponding to a 6.65% and 46.41% reduction in the feature space for LEX and LEM, respectively, compared to DLEX in Arabic. For English the number of lexemes is significantly smaller for the same data collection size, 41317 lexemes corresponding to 32890 lemmas, representing a relatively smaller reduction in the feature space, going from LEX to LEM, of 20.4%.

---

[3] It should be noted that lemmas are also lexemes however they are a specific inflectional form that are conventionally chosen as a citation form, for example a typical lemma for a noun is the inflected 3rd person masculine singular form of the noun.

[4] All the Arabic used in this paper uses the Buckwalter transliteration scheme as described in http://www.qamus.com

2.1.1.1   Extended Dimensions In order to reduce the sparseness of the NFR matrix and add a level of abstraction, we augment the features space with part of speech (POS) tag features. Algorithm 1 explains the mechanism of generating the congregated POS features.

Data: ANEAR window_size x, POS window_size y^5, input dataset
for *every PNE per $\in$ text_win* **do**
    if *distance(token, per)$\leq y$* **then**
        $tags = tags \bigcup POS\_tag(token)$
    end
end
for *every tag$\in$ tags* **do**
    increment the frequency$_{features\_vector(per)}$(class representative of POS tag)
end

Algorithm 1. Generate POS features

2.1.1.2   Feature Values are assigned based on one of the following metrics:

1. Co-occurrence Frequency (COF): PNE-feature co-occurrence frequency within a predetermined context window size of a sentence, SENT where the feature and the PNE co-occur in the same sentence, or a document, DOC, where the feature and the PNE co-occur in the same document. This results in either COF-SENT or COF-DOC.

2. Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF is calculated over the entire document collection. We have two settings varying the document size parameter for TF-IDF: (i) TF-IDF-DOC is based on using the entire collection of documents, and (ii) TF-IDF-PNE is based on constraining the document collection to those documents that mention the PNE. Both TF-IDF-DOC and TF-IDF-NE use the same equations as defined in 2 and 1 for calculating the feature values, however the former uses the entire document collection to calculate the values for DOC in the equations, while the latter is constrained to the document collection that mentions the PNE of interest, i.e. the vector row entry PNE in the matrix. Intuitively, both metrics capture the relative importance of the feature with respect to the PNE in a given document collection.

$$idf(feature, {}^6DOC) = log \frac{|DOCs|}{DOC \in DOCs : feature \in DOC} \qquad (1)$$

$$tf(feature, DOC) = \frac{feature\_count(feature, DOC)}{max\{feature\_count(feature, DOC) : feature \in DOC\}} \qquad (2)$$

3. Relative Rank Order (RRO): In this metric for the feature values, we abstract away from the absolute magnitude of the COF values or the TF-IDF values and we

Table 1. Sample NFR matrix illustrating the Feature Value (FV) Metrics COF-DOC values and their corresponding RRO values for the the various PNEs across 6 Lemma feature dimensions

	FV Metric	president	chief	kill	assassin	Saudi	negotiation
George Bush	*COF-DOC*	10	20	25	15	8	0
	RRO	4	2	1	3	5	0
Abu Ammar	*COF-DOC*	25	12	0	12	0	20
	RRO	1	3	0	3	0	2
Mahmoud Abbas	*COF-DOC*	20	11	8	1	0	35
	RRO	2	3	4	5	0	1
Abou Mazen	*COF-DOC*	24	16	5	2	0	30
	RRO	2	3	4	5	0	1
Yasser Arafat	*COF-DOC*	16	9	4	9	2	25
	RRO	2	3	4	3	5	1
G. W. Bush	*COF-DOC*	7	18	22	12	9	1
	RRO	5	2	1	3	4	6

replace them with their relative vector rank order value. Table 1 illustrates an example of the mapping between the COF-DOC values and the corresponding RRO values.[7]

2.1.2 Clustering and Retrieving the Different Groups of PNEs.

We apply unsupervised clustering using the cosine similarity function across the feature vectors in order to produce the multiple groups of entities along with their aliases, i.e. grouping PNEs. Our chosen clustering approach takes as input the NFR sparse matrix and applies the Repeated Bisection clustering method that locally and globally optimizes the clustering solution C which contains multiple groups of entities conjoined with their instances.

$$C = \left\{ c : \ c = \bigcup_{PNE\ e} alias_e \right\} \tag{3}$$

3 Evaluation

3.1 Data and Preprocessing Tools

All of our experiments use the GALE Phase (2) Release (1) parallel dataset for English & Arabic.[8] We preprocessed the Arabic and English datasets in order to produce the NER tags, lexemes, lemmas and the Arabic diacritized lemmas. For all the

---

[7] We experiment with assigning a rank order value of 0 to the features that have a COF/TFIDF value of 0 versus, giving it the lowest rank order value in a given vector. We note that assigning missing features a value of 0 yielded significantly better results over ranking the missing features as the lowest rank order in the vector due to two factors: assigning the 0 features the lowest rank renders the actual rank variable across different vectors introducing significant noise, i.e. similar missing features will have different rank order values across different PNE row entries. The effect is exacerbated given the significant sparseness in the matrix.

[8] LDC2007E103. (http://www.ldc.upenn.edu).

English preprocessing we use the Stanford CoreNLP toolset [1], for Arabic we use AMIRA by [2] for lexeme, diacritized lemma and undiacritized lemma generation. We use NIDA-ANER, the Arabic Named Entity Recognition by [3] to produce PNE tagged data. Figure 2 depicts the ANEAR processing steps.
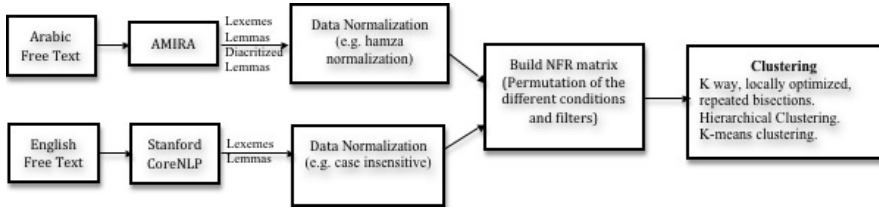

Fig. 2. ANEAR System Process

Due to the lack of annotated evaluation data for the aliasing resolution problem in Arabic and the limited evaluation data in English, we create our own English and Arabic evaluation data from the GALE dataset. Building the gold file comprises the following steps: a) Extract and list all the PNEs in the GALE dataset; b) In order to avoid singleton cases we set a unigram frequency threshold of ≥ 100 for each of the PNEs in order to be added to any of our clusters. This process yields an A list; c) Then we extract the transliterations of the PNEs based on string edit distance similarity measures for A; d) We then manually identify the aliases of the PNE in A in the dataset. The resulting gold standard file yields 26 PNE clusters in each language along with their respective aliases. The total number of PNEs in the Arabic set is 116 corresponding to 26 PNE clusters, and the total number of PNEs in English is 105 corresponding to 26 PNE clusters.

For automatic clustering, we use the CLUTO software package,[9] which employs multiple classes of k-way clustering algorithms that clusters low and high dimensional datasets with various similarity functions. CLUTO shows a robust clustering performance that outperforms many clustering algorithms such as K-means. We use the Repeated Bisection algorithm with default parameter settings. This clustering algorithm is a hard clustering algorithm. For clustering performance comparative reasons, we also use Matlab[10] implementations of the K-means and Hierarchical clustering algorithms.

3.2 Experimental Conditions

For each language, we have combinations of the following considerations. For the feature dimensions: (i) word tokenization level: Lexemes (LEX) vs. lemmas (LEM) vs. diacritized lexemes (DLEX) (the latter is only for Arabic). For the feature values, we have the following conditions: (i) simple co-occurrence frequency: COF-SENT and COF-DOC; (ii) TF-IDF-DOC and TF-IDF-NE; (iii) Rank Order with four settings:

---

[9] `http://glaros.dtc.umn.edu/gkhome/views/CLUTO`

[10] MATLAB and Statistics Toolbox Release 2009, The MathWorks, Inc., Natick, Massachusetts, United States.

RRO-COF-SENT, RRO-COF-DOC, and RRO-TFIDF-DOC, RRO-TFIDF-NE. We also have two feature sets: default bag of words, BOW, and BOW augmented with POS tag features, BOW+POS. Hence for English, this yields 2 word tokenization levels LEX/LEM * 8 feature value settings COF-SENT/COF-DOC/TF-IDF-DOC/TF-IDF-NE/RRO-COF-SENT/RRO-COF-DOC/RRO-TFIDF-DOC/RRO-TFIDF-NE amounting to 16 experimental conditions for each of the two feature settings BOW and BOW+POS, respectively. For Arabic, we have the following experimental conditions: 3 word tokenization levels LEX/LEM/DLEX *8 feature value settings COF-SENT/COF-DOC/TF-IDF-DOC/TF-IDF-NE/RRO-COF-SENT/RRO-COF-DOC/RRO-TF-IDF-DOC/RRO-TF-IDF-NE amounting to 24 experimental conditions for each of the two feature settings BOW and BOW+POS, respectively. Finally, we include the results of a naive baseline where the names are randomly assigned to one of 26 possible clusters, similar to our formulation of the problem, a PNE can only be assigned to one cluster (hard clustering).

3.3   Results

In Table 2, all the ANEAR conditions outperform the random baseline by a significant margin. ANEAR best results for English are obtained in the LEM_COF-DOC experimental setting achieving an $F_{\beta=1}$ score of 67.85% using the augmented POS features, and the best results for Arabic are achieved in the condition LEM_TF-IDF-DOC in the BOW+POS condition achieving an $F_{\beta=1}$=70.03%, with a narrow second condition LEX_TF-IDF-DOC with a score of $F_{\beta=1}$=69.58%.

In general with the BOW setting, the TF-IDF conditions outperform the comparative COF conditions. For example, in the English results, we note that LEX_TF-IDF-DOC|NE both outperform LEX_COF-SENT|DOC conditions (60.63% and 53.57% vs. 49.66% and 41.56%, respectively). Moreover, in the BOW setting, using RRO adversely impacts performance in both languages.

For both languages, The COF-DOC conditions outperform the COF-SENT conditions across the board. Also the TF-IDF-DOC conditions outperform the TF-IDF-NE conditions in the BOW setting, suggesting that narrowing the document collection extent is adverse to system performance.

For English, LEM conditions outperform LEX conditions except in the TF-IDF-DOC condition. However in the latter condition the difference between LEM and LEX conditions is relatively small (1%). In Arabic, the results are more consistent with LEM outperforming both LEX and DLEX in all the conditions, in the BOW setting.

Adding POS tag features has an overall positive impact on performance in English. In Arabic the story is quite different. The COF-SENT conditions in Arabic yield the worst results. But adding POS tag information to the other models seems to significantly improve performance.

For the Arabic experiments, under the BOW setting, the best F-score of 68.99% is obtained from the diacritized dataset (LEM) with TF-IDF-DOC. Using DOC provides better performance compared to SENT. Similarly to English results, adding POS tags to the feature space improves performance in both the LEX and LEM conditions, but not in the DLEX condition. This may be attributed to level of detail present in the DLEX forms combined with the detailed POS tag used. The best performing condition

Table 2. ANEAR $F_{\beta=1}$ scores performance for both English and Arabic datasets under the different experimental conditions and feature settings, BOW and BOW+POS

Condition	English		Arabic	
	BOW	BOW+POS	BOW	BOW+POS
Random Baseline	31.96	31.96	31.16	31.16
LEX_COF-SENT	41.56	44.19	56.52	42.4
LEX_RRO-COF-SENT	39.46	45.18	54.43	43.25
LEM_COF-SENT	43.05	39.84	60.17	39.36
LEM_RRO-COF-SENT	43.99	46.22	53.14	42.93
DLEX_COF-SENT	-	-	60.29	42.9
DLEX_RRO-COF-SENT	-	-	52.66	39.44
LEX_COF-DOC	49.66	64.25	59.15	62.75
LEX_RRO-COF-DOC	47.88	65.01	57.33	60.33
LEM_COF-DOC	51.91	**67.85**	64.42	62.75
LEM_RRO-COF-DOC	48.17	66.52	56.77	60.87
DLEX_COF-DOC	-	-	65.83	63.16
DLEX_RRO-COF-DOC	-	-	56.94	63.28
LEX_TF-IDF-NE	53.67	65.12	60.66	64.25
LEX_RRO-TF-IDF-NE	46.55	63.82	53.51	65.64
LEM_TF-IDF-NE	57.41	64.36	67.3	65.83
LEM_RRO-TF-IDF-NE	47.09	63.82	49.93	60.87
DLEX_TF-IDF-NE	-	-	66.63	65.83
DLEX_RRO-TF-IDF-NE	-	-	49.45	60.87
LEX_TF-IDF-DOC	60.63	64.47	65.88	69.58
LEX_RRO-TF-IDF-DOC	36.05	62.62	40.26	63.12
LEM_TF-IDF-DOC	59.65	62.67	65.12	**70.03**
LEM_RRO-TF-IDF-DOC	40.52	62.74	40.6	62.08
DLEX_TF-IDF-DOC	-	-	68.99	66.76
DLEX_RRO-TF-IDF-DOC	-	-	41.19	62.08

yields an f-score of 70.03% in the LEM, TF-IDF-DOC setting. This is a significant improvement over the same condition setting without POS tag features which yielded an f-score of 65.12% only. It is worth noting that the POS tag set in Arabic is quite rich almost fully specifying the morphology of the word encoding significant semantic attributes unlike the English tag set that is purely syntactic. The emphasis on semantic features seems to be further corroborated by the noticeable improvement using LEM compared DLEX and LEX, leading to a more dense representation. Moreover more evidence comes from the fact that DLEX outperforms LEX in all the DOC conditions.

4 Discussion

4.1 Balancing the Data

We are cognizant of the unbalanced distribution of the aliases in the dataset within one cluster which highly affects the clustering performance. Hence, in addition to testing on

Data: The free text, Gold clusters
Result: A new redistribution of gold PNEs in the input text
for *every occurrence of a PNE **name** that is in the gold clusters* **do**

> cluster_id = get cluster ID of the input name
> with the use of uniformly distributed random number generator, retrieve randomly a member $new\_alias : new\_alias \in cluster_{cluster\_id}$
> replace $name$ with new_alias

end

Algorithm 2. Balancing and Resampling the dataset

the original dataset, we generate another balanced version that has a more normalized distribution based on the following approach:

When we balance the evaluation data, we observe an overall significant increase in absolute performance where the best condition LEM_COF-SENT yields an F-score of 96.05% for English compared to the best condition in Arabic of LEM_TFIDF-NE yielding an F-score of 96.45%.
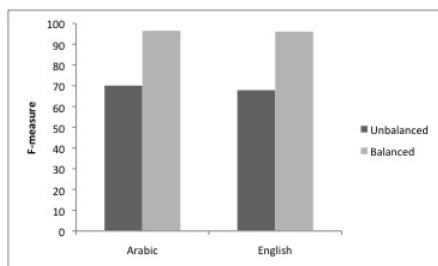
Fig. 3. ANEAR performance comparison between balanced and unbalanced Arabic and English datasets
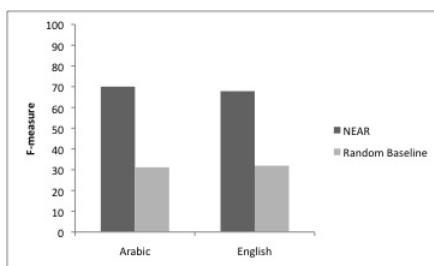

Fig. 4. Comparison between ANEAR and random baseline performance

Arabic shows more robust results and seems less affected (f-score = 70.03%) when compared to English (f-score = 67.85%). The more balanced distribution scheme adds a significant performance improvement ($\approx +25\%$) as shown in Figure 3. Based on the results, we generally notice that diacritized lexemes produce better performance, despite the higher feature dimensionality that yields a more sparse data set, yet decreasing the ambiguity results is a gain. Figure 3 contrasts ANEAR performance against a random baseline system with a gain of $\approx +39\%$ in Arabic and $\approx +30\%$ in English.

4.2 Alternate Clustering Algorithms

Additionally, we carry out a comparison assessment evaluation for our system against different clustering algorithms, namely, K-Means and Hierarchical clustering. Both K-Means and CLUTO Repeated Bisection require the number of clusters as an input parameter, and they yield their best performance under the same conditions. Whereas

Hierarchical clustering, though it does not require specifying the number of clusters as an input parameter, the number of clusters is automatically induced, it yields much poorer F-score results.

K-Means achieves the best performance under the condition DLEX_TF-IDF-NE (in Arabic) with an $F_{\beta=1}$ score of 36.49%. On the other hand, Hierarchical clustering shows its best performance under the condition: LEX_COF-DOC with an $F_{\beta=1}$ score of 21.38%. Figure5 shows a comparison among the different clustering algorithms when tested on balanced and unbalanced dataset.
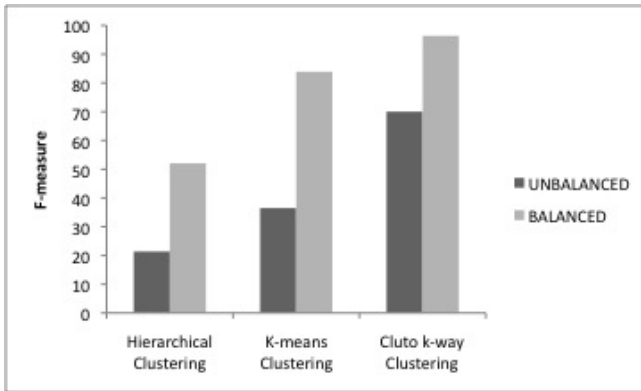

Fig. 5. Comparison among Hierarchical, K-means and CLUTO Repeated Bisection K-way Clustering when tested on the Arabic balanced and unbalanced datasets

5   Related Work

To date, most of the work related to the aliasing resolution problem has been mainly performed in the area of Named Entity Disambiguation, where two entities share the same name. Moreover, the NED task has typically focused on English since there are no annotated data sets for other languages. Our work employs unsupervised techniques to induce the PNE groups of name aliases while most work that we are aware of to date, uses predefined lists of PNEs and their corresponding aliases and used for training in a supervised manner. [4] proposed a framework for alias detection for a given entity using a logistic regression classifier that relies on a number of features such as co-occurrence relevance. Similarly, [5] presented a more complicated system that also relies on an input list of names and their aliases. They first retrieve a list of candidate aliases for a given entity using lexical patterns that introduce aliases, then they rank the set of retrieved aliases based on different factors: a) Lexical pattern frequency, b) Co-occurrence in anchor texts using different metrics such as TF-IDF and cosine similarity functions, and, c) Page counts of name-alias co-occurrence. [6,7] and [8] proposed a knowledge-based method that captures and leverages the structural semantic knowledge in multiple knowledge sources (such as Wikipedia and WordNet) in order to improve the disambiguation performance. Other disambiguation methods utilize ranked similarity measurements among entity-based summaries. [9,10]. [11] have used unsupervised

clustering algorithms on a rich feature space that is extracted from biographical facts. In PNE identification, [12] proposes a lexical pattern-based approach to extract a large set of candidate aliases from a web search engine. Then, a myriad of ranking scores (lexical pattern frequency, word co-occurrences and page counts on the web) are integrated into a single ranking function and fed into a support vector machines (SVM) to identify and predict aliases for a particular PNE.

Other contributions involved handling structured datasets such as Link Data Sets. [13] presented a hybrid probabilistic orthographic-semantic supervised learning model to recognize aliases.

Entity linking tackles a similar problem to NEAR where a name mention is mapped to an entry in a Knowledge Base (KB). Entity Linking relies heavily on Wikipedia pages to populate the KB and generates a dictionary that is used in name-variant mappings as illustrated in [14]. They integrate a number of features in order to choose the best mapping. These features include the surface forms, semantic links which assumes the availability of structured data and weighted bag of words features that are extracted from the Wikipedia documents. All of the above features assume that the entities to be resolved with their aliases are celebrities where Wikipedia reference them and their aliases.

Our approach provides a broader range of alias identification, since it does not rely on any lexical or string similarity properties. In addition, the identification process is executed offline with no dependence on external resources.

6   Conclusion

In this paper, we present a statistical, domain-independent aliasing resolution system, ANEAR. In building our system and exploring the search space, we experiment with different feature types and values and we measure their impact within two different languages Arabic and English. We note that employing semantically and syntactically oriented features helps performance. Also our results suggest that balancing the data set, namely the alias distribution, plays a role in improving performance. Our system is the first for ANEAR in Arabic. Our work results in annotated data sets for both Arabic and English.

Our best results on unbalanced Arabic and English datasets are $F_{\beta=1} = 70.03\%$ and $F_{\beta=1} = 67.85$, respectively.

References

1. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL 2005, pp. 363–370. Association for Computational Linguistics, Stroudsburg (2005)

2. Diab, M.: Second generation tools (amira 2.0): Fast and robust tokenization, pos tagging, and base phrase chunking. In: Choukri, K., Maegaard, B., eds.: Proceedings of the Second International Conference on Arabic Language Resources and Tools. The MEDAR Consortium, Cairo (2009)

3. Benajiba, Y., Diab, M.T., Rosso, P.: Arabic named entity recognition: A feature-driven study. IEEE Transactions on Audio, Speech & Language Processing 17(5), 926–934 (2009)

4. Jiang, L., Wang, J., Luo, P., An, N., Wang, M.: Towards alias detection without string similarity: an active learning based approach. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2012, pp. 1155–1156. ACM, New York (2012)

5. Bollegala, D., Matsuo, Y., Ishizuka, M.: Automatic discovery of personal name aliases from the web. IEEE Trans. on Knowl. and Data Eng. 23(6), 831–844 (2011)

6. Han, X., Zhao, J.: Structural semantic relatedness: A knowledge-based method to named entity disambiguation. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 50–59. Association for Computational Linguistics, Uppsala (2010)

7. Cucerzan, S.: Large-scale named entity disambiguation based on wikipedia data. In: Proceedings of EMNLP-CoNLL, vol. 2007, pp. 708–716 (2007)

8. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: IJCAI 2007: Proceedings of the 20th International Joint Conference on Artifical Intelligence, pp. 1606–1611. Morgan Kaufmann Publishers Inc., San Francisco (2007)

9. Bagga, A., Baldwin, B.: Entity-based cross-document coreferencing using the vector space model. In: COLING-ACL, pp. 79–85 (1998)

10. Bagga, A., Biermann, A.W.: A methodology for cross-document coreference. In: Proceedings of the Fifth Joint Conference on Information Sciences (JCIS 2000), pp. 207–210 (2000)

11. Mann, G.S., Yarowsky, D.: Unsupervised personal name disambiguation. In: Daelemans, W., Osborne, M. (eds.) Proceedings of CoNLL-2003, pp. 33–40. Edmonton, Canada (2003)

12. Bollegala, D., Matsuo, Y., Ishizuka, M.: Automatic discovery of personal name aliases from the web. IEEE Trans. Knowl. Data Eng. 23(6), 831–844 (2011)

13. Hsiung, P., Moore, A., Neil, D., Schneider, J.: Alias detection in link data sets. Master's thesis, Technical Report CMU-RI-TR-04-22 (March 2004)

14. Charton, E., Gagnon, M.: A disambiguation resource extracted from wikipedia for semantic annotation. In: LREC, pp. 3665–3671 (2012)

15. Chen, Y., Martin, J.: Towards robust unsupervised personal name disambiguation. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 190–198. Association for Computational Linguistics, Prague (2007)

16. Sutton, C., Mccallum, A.: Introduction to Conditional Random Fields for Relational Learning. MIT Press (2006)

Improving Candidate Generation for Entity Linking

Yuhang Guo[1], Bing Qin[1,*], Yuqin Li[2], Ting Liu[1], and Sheng Li[1]

[1] School of Computer Science and Technology,
Harbin Institute of Technology, Harbin, China
[2] Beijing Information Science and Technology University, Beijing, China
`{yhguo,bqin,tliu,sli}@ir.hit.edu.cn,`
`li.yuqin@trs.com.cn`

Abstract. Entity linking is the task of linking names in free text to
the referent entities in a knowledge base. Most recently proposed linking
systems can be broken down into two steps: candidate generation and
candidate ranking. The first step searches candidates from the knowl-
edge base and the second step disambiguates them. Previous works have
been focused on the recall of the generation because if the target entity
is absent in the candidate set, no ranking method can return the cor-
rect result. Most of the recall-driven generation strategies will increase
the number of the candidates. However, with large candidate sets, mem-
ory/time consuming systems are impractical for online applications. In
this paper, we propose a novel candidate generation approach to gen-
erate high recall candidate set with small size. Experimental results on
two KBP data sets show that the candidate generation recall achieves
more than 93%. By leveraging our approach, the candidate number is
reduced from hundreds to dozens, the system runtime is saved by 70.3%
and 76.6% over the baseline and the highest micro-averaged accuracy in
the evaluation is improved by 2.2% and 3.4%.

Keywords: Natural Language Processing, Information Extraction, En-
tity Linking, Candidate Generation, Candidate Pruning.

1 Introduction

Entity Linking (EL) is the task of identifying the target entity which a name
refers to. It can help text analysis systems to understand the context of the name
in-depth by leveraging known information of the entity. On the other hand, new
knowledge about this entity can be populated by mining information from the
context. Figure 1 illustrates entity linking can help question answering: knowing
the name *Washington* refers to actor `Denzel Washington` (rather than `George
Washington` or the `State of Washington`) in the question: *Who did Washington
play in Training Day*, one can find the corresponding answer (*Detective Alonzo
Harris*) directly in the knowledge base.

---

* Corresponding author.

Fig. 1. An example of entity linking

EL can be broken down into two steps: candidate generation and candidate ranking. The first step generates a set of candidate entities of the target name and the second step ranks the candidates. Several ranking models have been proposed for the second step. However, few works have focused on the candidate generation step. Generating candidates is a critical step for the linking systems. If the target entity is not included in the candidate set, no ranking model can return the correct one.

A number of resources have been proposed to improve the generation recall [2,4,24,6]. By leveraging these resources, the number of the candidates can be very big. Take the target name *Washington* for example, the generation will return more than 600 candidates.

Bounding the number of the candidates is important in the applications of EL. Lessening the candidates will reduce time and memory costs of the ranking, and further make sophisticated time and memory consuming ranking models be practicable. How to generate small candidate sets under the premise of ensuring high recall is an interesting problem.

In this paper, we propose a novel candidate generation approach. In this approach, the generator first extracts the target name's co-reference names in the context. From this set the generator then selects the most reliable name (i.e. the least ambiguous name) to generate candidates by leveraging a Wikipedia-derived name-entity mapping. Next the generator prunes the candidates according to their frequencies and their similarity to the target name.

Experiment on benchmark data sets shows that our candidate generation can increase the recall and reduce the candidate number effectively. Further analysis shows that both the accuracy and the speed of the system can benefit from the proposed candidate generation approach, especially for the target names with large candidate set. The system runtime can be effectively saved over the baseline

candidate set. The highest accuracy in the evaluation is improved by 2.2% and 3.4%.

2   Related Work

EL is similar to Word Sense Disambiguation (WSD), a widely-studied natural language processing task. In WSD the sense of a word (e.g. bank: river bank or a financial institution) is identified according to the context of the word [10,20,15]. Both WSD and EL disambiguate polysemous words/names according to the context. The difference between the two tasks is in that, the disambiguation targets in WSD are lexical words whereas in EL are names. In WSD, the senses of words are defined in dictionaries, such as WordNet [18]. In EL, however, no open domain catalog has included all entities and all of their names. The study on WSD have a history of several decades[10,20,15]. Recently, as the development of the large scale open domain knowledge bases (such as Wikipedia, DBpedia[1,1] and Yago[23], etc.), EL has been attracting more and more attentions.

Early EL borrowed successful techniques in WSD: take each sense (candidate entity) as a class and resolve the problem by multi-class classifier[17,2]. However, in WSD a word usually has several senses but in EL a name may have dozens to hundreds of candidate entities. Under such high polysemy, the accuracy of the classifier cannot be guaranteed.

EL systems can be broken down into two steps: candidate generation and candidate ranking[11]. Early candidate generation approaches directly match the target name in the knowledge base[2]. Recently, several techniques have been proposed and have achieved certain success in recall.

- Substitute the target name to a longer name in the names co-reference chain in the context[4].
- If the target name is an acronym, substitute it with the full name in the context[4,24].
- Filter acronym expansions with a classifier[26].
- If the exact match fails, then use partial search[24] or fuzzy match[14] (e.g. return candidates with high Dice coefficient).

The candidate ranking is based on the similarity between the candidate entity and the context surround with the target name. A number of features have been proposed: Plain text[24]; Concepts, such as Wikipedia category[2], Wikipedia concept[9], topic model concept[12,22,26]; And neighboring entities, which include the entities mapped from unambiguous names[19] and the collectively disambiguated entities[4,13,8,22]. The entity-context similarity is measured in: cosine similarity[24], language model score[7] and the inner coherence among neighboring entities measured by link similarity[19,21,22] and collective topic model similarity[22]. Besides directly use these similarities for the ranking, machine learning methods has been applied to combine these similarities[19,27,5].

Sophisticated ranking models need heavy computation costs. For example, the time complexity of the list wise learning to rank method is exponential[3,25,27].

Collective disambiguation is NP hard[13]. Therefore, generating small candidate sets is important to these ranking models. However, little work to date has focused on the candidate pruning.

3   Candidate Generation Approach

An entity may be mentioned many times with different names in document. Some of the names are easier to be linked than others. For example, *Denzel Washington* is less ambiguous than *Washington*. In this paper we propose a context based candidate generation approach (CBCG). CBCG first detects co-reference names of the target name in the context. The co-reference names are the target entity's potential names, including acronym expansion, longer names and shorter names. Then the approach match probably the least ambiguous potential name in a Wikipedia-derived Name-Entity Mapping (NEM). Next the returned entities are filtered by their frequency and their similarity to the target name. To summarize, CBCG reduces the candidate number by leveraging three strategies: back-off, filter by frequency, and filter by similarity.

Using the back-off strategy, the most reliable name is first considered. The CBCG generator considers the next most reliable name only if the current name returns no candidate. Using the filter by frequency strategy, the generator set a volume threshold to the candidate set and low frequency candidate will be filtered. Using the filter by similarity strategy, the generator set a similarity threshold and the candidates with low similarity with the target name will be filtered. In the following of this section, we will describe the NEM construction, the potential name detection, and the candidate pruning in detail.

3.1   Name-Entity Mapping Construction

An entity may be mentioned in different name. Some of these name variations (or aliases, alternative names) represent the entity frequently, and some others not. Collecting as many name variations of the entity as possible involves the recall when this entity is referred to. Name-entity pairs and the co-occurrence frequency can be mined from the following Wikipedia structure:

- Page and redirect page title of the entity.
- Title of the disambiguation page which contains the entity.
- Anchor text which targets to the entity.
- Bold text in the first paragraph of the entity.
- Value of the name field (e.g. birth_name, nick_name, etc.) within Infobox,[1].

In this mapping, a name is mapped to all the entities it may refer to. For example, name *Washington* is mapped to `Denzel Washington`, `George Washington` and `State of Washington`, etc. All through this work, we use the Aug. 2, 2012

---

[1] A information structure of Wikipedia.

version of English Wikipedia dump, which contains more than 4.1 million articles[2]. In all, we extract 23,895,819 name-entity pairs with their co-occurrence frequencies. Summing up this frequency for the same entity, we can get the frequency of the entity in Wikipedia, which will be used in the following part of the linking system.

3.2 Potential Name Detection

We first apply forward maximum match algorithm on the context to extract all names that match in the NEM. Then we select the names which contains the target name (i.e. longer name) or is a substring of the target name (i.e. shorter name) as the potential name of the target entity.

Besides longer names and shorter names, potential name set also contains transformations of the target name. Because many all-capital names (e.g. *AR-GENTINA*) cannot be matched in the NEM, we normalize the non-acronym all-capital words into Wiki-style[3]. We also substitute the state abbreviation names[4] (e.g. *CA*) in the document into the full forms (e.g. *California*).

Acronym target names (i.e. *ABC*) should be considered separately. The acronyms and their full names usually satisfy other constrains. For example,

- The full form is in front of the enclosed acronym (e.g. ... the newly formed All Basotho Convention (ABC))
- The acronym is in front of the enclosed full form (e.g. ... at a time when the CCP (Chinese Communist Party) claims ...)
- The acronym consists of the initial letters of the full name words (e.g. ... leaders of Merkel's Christian Democratic Union ... CDU ...)

These cases can be covered by several regular expressions.

Here we propose a novel acronym identification rule: a name string is an acronym if it satisfies all the following conditions:

- It contains no more than 4 letters.
- It contains no less than 2 upper case letters.
- It contains no more than 2 lower case letters.

According to the above rules, for example, *ABC* and *MoD* are identified as acronyms, and *Abbott* and *ARGENTINA* are not.

3.3 Candidate Pruning

The objective of this step is to minimize the set size of the candidates to be generated and maximize the possibility that the target entity is reserved in the set. The first strategy to reduce the number of the candidates is the back-off strategy: consult the most reliable potential name for the generation and

consult the next most reliable potential name only if the current name returns no candidate.

Two points should be considered for the reliability of the potential name:

1. The number of the entities generated by this name. (N)
2. The probability of this potential name is a name of the target entity.(P)

According to our observation, longer name has a smaller N and higher frequency name has a higher P. In order to keep a small candidate set and a high recall at the same time, the considered potential name should be of both high frequency and long.

In this work, the potential names are first sorted by their types: longer names, normalized query names (including acronym expansion and Wiki-style normalization) and shorter names, and then by frequency in the same type.

The back-off strategy prunes candidates from name aspect. Whereas the following strategies prune candidates from the entity aspect. The filter by frequency strategy filter out the candidates with low frequency and the filter by similarity strategy filter out candidates with low similarity to the target name. We define the similarity between a name and an entity as follows: The target name n_t is similar to a candidate entity e if and only if at least one name (n_e) of this entity is similar to n_t.

Here we propose a novel name similarity measurement. The formula is

$$Sim(n_e, n_t) = \frac{\sum_{w \in n_e} Len(LCS(w, n_t))}{\sum_{w \in n_t} Len(w)} \tag{1}$$

where $Len(s)$ is the length of string s, $LCS(s_1, s_2)$ is the longest common string of s_1 and s_2. Note that this similarity is asymmetric.

Table 1. Notations

$Sort(\cdot)$	Returns a queue sorted by frequency.
$Pop(\cdot)$	Pop the top element in a queue.
$Sim(s_1, s_2)$	Return the similarity between string s_1, s_2.
$E(n)$	Returns the entities whose name matches n.
$N(e)$	Returns the names whose entity matches e.
n_t	Target name.
e_t	Target entity
N_p	Potential name set of e_t
C	Candidate entity set
Q_N	Potential name queue, sorted by frequency.
Q_E	Entity queue, sorted by frequency.

The candidate pruning strategies are combined in Algorithm 1. The symbols are described in Table 1 The candidate number is controlled by two parameters: a

similarity threshold is used to filter out the un-similar entities, and the candidate set volume threshold limits the maximum size of the candidate set[5].

---

input : target name n_t, potential name queue Q_N, similarity threshold p, and candidate set volume threshold T

output: candidate set C

1 $C \leftarrow \phi$;
2 **while** $C = \phi$ *and* $Q_N \neq \phi$ **do**
3 $\quad$ $n \leftarrow \texttt{Pop}(Q_N)$;
4 $\quad$ $Q_E \leftarrow \texttt{Sort}(\texttt{E}(n))$;
5 $\quad$ $i \leftarrow 0$;
6 $\quad$ **while** $Q_E \neq \phi$ *and* $i < T$ **do**
7 $\quad\quad$ $i \leftarrow i + 1$;
8 $\quad\quad$ $e \leftarrow \texttt{Pop}(Q_E)$;
9 $\quad\quad$ **for** $n_e \in \texttt{N}(e)$ **do**
10 $\quad\quad\quad$ **if** $\texttt{Sim}(n_e, n_t) > p$ **then**
11 $\quad\quad\quad\quad$ $C \leftarrow C \cup \{e\}$;
12 $\quad\quad\quad\quad$ break;
13 $\quad\quad\quad$ **end**
14 $\quad\quad$ **end**
15 $\quad$ **end**
16 **end**

Algorithm 1. Candidate generation and pruning

4   Experiment

The experiment is conducted on four KBP data sets (i.e. KBP2009-KBP2012) which are taken from the Knowledge Base Population (KBP) Track [16,11]. The data sets share the same track knowledge base which is derived from Wikipedia and contains 818,741 entities. We use KBP2009 and KBP2010 as the training and development data and KBP2011 and KBP2012 as the test data.

In the KBP-EL evaluation, the input is a set of queries. Each query consists of a target name mention and a context document. The output is the target entity ID in the knowledge base or NIL if the target entity is absent in the knowledge base. The number of queries/NIL-answer queries for each data set is: KBP2009: 3904/2229, KBP2010: 2250/1230, KBP2011: 2250/1126, KBP2012: 2250/1049.

Our experiments include two parts. The first part evaluates the recall and averaged candidate set size. The recall is the percentage of the non-NIL queries for which the candidate set covers the referent entity. The second part evaluates the final EL system performance, including the micro-averaged accuracy (percentage of queries linked correctly) and the averaged runtime cost per query.

---

[5] In this work we set the candidate set volume threshold 30 and the similarity threshold 0.6.

4.1   Evaluation on Recall

Here we compare our context based approach: CBCG with the baseline, directly matching in NEM: DMatch. Table 2 shows the recall and the averaged candidate number per query of the candidate generators. From this table, we can see that the recall of CBCG outperforms DMatch and can achieve higher than 93% on each of the data sets. On KBP2011 and KBP2012, the recall of CBCG outperforms DMatch by 15.6% and 5.2% respectively. On the other hand, the number of the candidates of CBCG only 22.5% and 9.5% of DMatch on KBP2011 and KBP2012 respectively. Few literature has reported both of the recall and the averaged candidate number. The Literature [6] reported their candidate generation recall was 0.878 and the averaged candidate number was 7.2 on KBP2009. Our approach outperforms the recall by 5.3% achieves a comparable candidate number on the same data set.

Table 2. Candidate generation recall and averaged candidate number on KBP data sets

Data Set	KBP2009	KBP2010	KBP2011	KBP2012
Recall				
DMatch	0.906	0.900	0.807	0.883
CBCG	0.931	0.964	0.963	0.935
Averaged Candidate Number				
DMatch	24.6	28.5	38.3	132.3
CBCG	8.0	8.5	8.6	12.6

The CBCG can be broken down into the following strategies:

DMatch: Directly match the target name in NEM
AcroExp: Add acronym expansion into the potential name set
LongName: Add longer co-reference of the target name into the potential name set
ShortName: Add shorter co-reference of the target name into the potential name set
fByFreq: filter by candidate frequency
fBySim: filter by candidate similarity with the target name

We add the strategies into the generator in turn to evaluate their contributions. Figure 2 shows that, directly matching the target name in NEM results in a large number of candidates. Using AcroExp, LongName and ShortName strategies, the recall will be improved. Using LongName and fByFreq, the averaged number of the candidates will be reduced significantly. Using all of these strategies, we can obtain balanced candidate sets with high recall and small size.
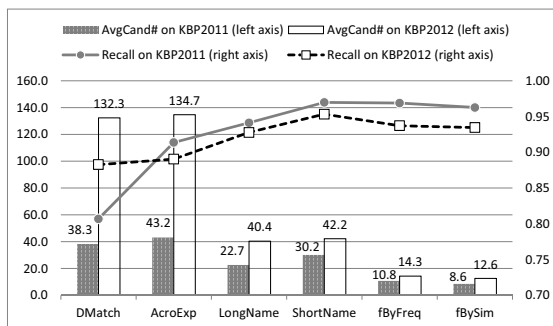
Fig. 2. Recall and averaged candidate number by different strategies

4.2 Evaluation on Accuracy

We use three candidate ranking models to evaluate the final performance of the EL system. The ranking is based on the results of our candidates generation. The ranking models are: (1) The vector space model based on cosine similarity between the candidate and the context of the target name: VSM [4,24]; (2) The machine learning model based on list wise learning to rank: ListNet [27]; and (3) The language model: LM [7]. VSM method is a simple but effective ranking model. ListNet is a state-of-the-art ranking model. LM is also a state-of-the-art ranking method but is time and memory consuming. The system output NIL if the candidate set is empty or the top ranked entity is absent from the track knowledge base. We compare the systems with the top 3 systems in the KBP evaluation. VSM+DMatch, ListNet+DMatch and VSM+CBCG, ListNet+CBCG are based on the candidate set of the DMatch baseline and the CBCG respectively. The baseline (DMatch) generated so many candidates that the LM model ran out of memory in our machine. So currently we can not provide the result of the LM model based on DMatch.

From Table 3 we can see that, the accuracies of VSM+CBCG and List-Net+CBCG are significantly higher than their DMatch versions (improved by 5.4%-11.4%) respectively, and the accuracy of LM+CBCG outperforms the best systems in the evaluations by 2.2% and 3.4% on KBP2011 and KBP2012 respectively.

4.3 Evaluation on Efficiency

Figure 3 shows the query numbers in different candidate number (i.e. polysemy) ranges on KBP2011 and KBP2012. From this figure, we can see that on KBP2011 nearly a half of the queries have more than 16 candidates, and on KBP2012 over a half of the queries have more than 64 candidates. *Jackson* is the most polysemous target name in the data sets, which has 865 candidates. From Table 2 we can see that the averaged candidate number on KBP2012 is up to 132.3. Such a big number of candidate is impractical for online applications. So the candidate pruning is essential for the candidate generation.

Table 3. EL accuracy on KBP2011 and KBP2012

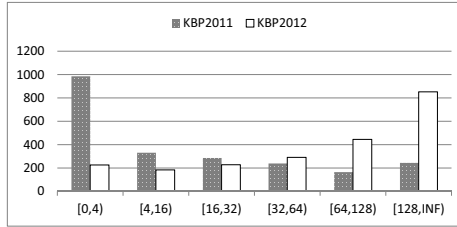| Data Set	KBP2011	KBP2012
Sys1	0.863	0.766
Sys2	0.861	0.757
Sys3	0.790	0.755
VSM+DMatch	0.689	0.558
VSM+CBCG	0.777	0.672
ListNet+DMatch	0.690	0.622
ListNet+CBCG	0.786	0.676
LM+CBCG	0.885	0.800

Fig. 3. Query numbers in different candidate number range

We use ListNet to evaluate the candidate generation efficiency. Figure 4 shows the runtime cost and the accuracy of the ListNet model on the baseline candidate set (DMatch) and the proposed candidate set (CBCG). From Figure 4 we can see that, the runtime increases significantly for the DMatch and keeps steady for the CBCG. By leveraging the proposed candidate set, the accuracies are improved in all candidate number ranges. For the most polysemous targets, the accuracy is improved by 18.0% and 8.6% and the runtime is saved by 90.2% and 85.8% on KBP2011 and KBP2012 respectively. In total, the accuracy is improved by 9.6% and 5.4% and the runtime of the system is saved by 70.3% and 76.6%.
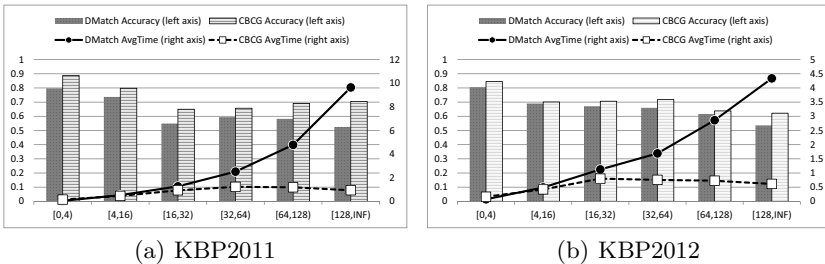

(a) KBP2011   (b) KBP2012

Fig. 4. Accuracy and averaged time cost per query (seconds) of ListNet based on the DMatch and the CBCG in different polysemy ranges on KBP2011 and KBP2012

5   Conclusion

Candidate generation is essential for the EL task. The candidate number for the target names may be very large. Generating small candidate set under the premise of ensuring high recall is critical for the applications of the EL systems. In this paper we propose a novel candidate generation approach. This approach combines several strategies to balance the recall and the size of the candidate set. Experimental results on benchmark data set shows that our candidate generation can significantly improve the EL system performances on recall, accuracy and efficiency over the baseline. On the KBP2011 and KBP2012 data sets, the recall is improved by 15.6% and 5.2%, the accuracy is improved by 5.4%-11.4%, the system runtime is saved by 70.3% and 76.6%, and the highest accuracy in the evaluation is improved by 2.2% and 5.4% respectively. For the most polysemous target names on KBP2011 and KBP2012, the accuracy improvement achieves 18.0% and 8.6%, and the runtime is saved by 90.2% and 85.8% respectively.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: A nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
2. Bunescu, R.C., Pasca, M.: Using encyclopedic knowledge for named entity disambiguation. In: EACL (2006)
3. Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the 24th International Conference on Machine Learning (2007)
4. Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (2007)
5. Dredze, M., McNamee, P., Rao, D., Gerber, A., Finin, T.: Entity disambiguation for knowledge base population. In: Proceedings of the 23rd International Conference on Computational Linguistics (2010)
6. Hachey, B., Radford, W., Nothman, J., Honnibal, M., Curran, J.R.: Evaluating entity linking with wikipedia. Artificial Intelligence 194, 130–150 (2013)
7. Han, X., Sun, L.: A generative entity-mention model for linking entities with knowledge base. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Techologies (2011)
8. Han, X., Sun, L., Zhao, J.: Collective entity linking in web text: a graph-based method. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information (2011)

9. Han, X., Zhao, J.: Named entity disambiguation by leveraging wikipedia semantic knowledge. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009 (2009)
10. Ide, N., Véronis, J.: Introduction to the special issue on word sense disambiguation: the state of the art. Comput. Linguist. 24(1), 2–40 (1998)
11. Ji, H., Grishman, R.: Knowledge base population: Successful approaches and challenges. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (2011)
12. Kataria, S.S., Kumar, K.S., Rastogi, R.R., Sen, P., Sengamedu, S.H.: Entity disambiguation with hierarchical topic models. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2011)
13. Kulkarni, S., Singh, A., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of wikipedia entities in web text. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2009)
14. Lehmann, J., Monahan, S., Nezda, L., Jung, A., Shi, Y.: Lcc approaches to knowledge base population at TAC 2010. In: Proceedings of the Text Analysis Conference (2010)
15. McCarthy, D.: Word sense disambiguation: An overview. Language and Linguistics Compass 3(2), 537–558 (2009)
16. McNamee, P., Dang, H.: Overview of the tac 2009 knowledge base population track. In: Proceedings of the Second Text Analysis Conference, TAC 2009 (2009)
17. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007 (2007)
18. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: Introduction to WordNet: An On-line Lexical Database*. Int. J. Lexicography 3, 235–244 (1990)
19. Milne, D., Witten, I.H.: Learning to link with wikipedia. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008 (2008)
20. Navigli, R.: Word sense disambiguation: A survey. ACM Comput. Surv. 41, 1–69 (2009)
21. Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (2011)
22. Sen, P.: Collective context-aware topic models for entity disambiguation. In: Proceedings of the 21st International Conference on World Wide Web (2012)
23. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web (2007)
24. Varma, V., Bharat, V., Kovelamudi, S., Bysani, P., Santhosh, G.S.K., Kiran Kumar, N., Reddy, K., Kumar, K., Maganti, N.: IIIT hyderabad at TAC 2009. In: Proceedings of the Second Text Analysis Conference, TAC 2009 (2009)
25. Xia, F., Liu, T.-Y., Wang, J., Zhang, W., Li, H.: Listwise approach to learning to rank: theory and algorithm. In: Proceedings of the 25th International Conference on Machine Learning (2008)
26. Zhang, W., Sim, Y.C., Su, J., Tan, C.L.: Entity linking with effective acronym expansion, instance selection, and topic modeling. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011, Barcelona, Catalonia, Spain, July 16-22 (2011)
27. Zheng, Z., Li, F., Huang, M., Zhu, X.: Learning to link entities with knowledge base. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (2010)

Person Name Recognition Using the Hybrid Approach

Mai Oudah[1] and Khaled Shaalan[1,2]

[1] The British University in Dubai, UAE
[2] School of Informatics University of Edinburgh, UK
`oudah.mai@gmail.com, khaled.shaalan@buid.ac.ae`

Abstract. Arabic Person Name Recognition has been tackled mostly using either of two approaches: a rule-based or Machine Learning (ML) based approach, with their strengths and weaknesses. In this paper, the problem of Arabic Person Name Recognition is tackled through integrating the two approaches together in a pipelined process to create a hybrid system with the aim of enhancing the overall performance of Person Name Recognition tasks. Extensive experiments are conducted using three different ML classifiers to evaluate the overall performance of the hybrid system. The empirical results indicate that the hybrid approach outperforms both the rule-based and the ML-based approaches. Moreover, our system outperforms the state-of-the-art of Arabic Person Name Recognition in terms of accuracy when applied to ANERcorp dataset, with precision 0.949, recall 0.942 and f-measure 0.945.

Keywords: Person Name Recognition, Natural Language Processing, Rule-based Approach, Machine Learning Approach, Hybrid Approach.

1 Introduction

Named Entity Recognition (NER) is the task of detecting and classifying proper names within texts into predefined types, such as Person, Location and Organization names [19], in addition to the detection of numerical expressions, such as date, time, and phone number. Many Natural Language Processing (NLP) applications employ NER as an important preprocessing step to enhance the overall performance.

Arabic is the official language in the Arab world where more than 300 million people speak Arabic as their native language [22]. Arabic is a Semitic language and one of the richest natural languages in the world in terms of morphology [22]. Interest in Arabic NLP has been gaining momentum in the past decade, and some of the tasks, such as NER, have proven to be challenging due to the language's rich morphology.

Person Name Recognition for Arabic has been receiving increasing attention, yet opportunities for improvement in performance are still available. Most of the Arabic NER systems, which have the capability of recognizing Person names, have been developed using two types of approaches: the rule-based approach, notably NERA system [24], and the ML-based approach, notably ANERsys 2.0 [6]. Arabic rule-based NER systems rely on handcrafted grammatical rules acquired from linguists. Therefore, any maintenance applied to rule-based systems is labor-intensive and time consuming especially if linguists with the required knowledge are not available [21].

On the contrary, ML-based NER systems utilize learning algorithms that make use of a selected set of features extracted from datasets annotated with named entities (NEs) for building predictive NER classifiers. The main advantages of the ML-based NER systems are that they are updatable with minimal time and effort as long as sufficiently large datasets are available.

In this paper, the problem of Arabic Person Name Recognition is tackled through integrating the ML-based approach with the rule-based approach to develop a hybrid system in an attempt to enhance the overall performance. Our early hybrid Arabic NER research [1] provided the capability to detect and classify Person NEs in Arabic texts in addition to Location and Organization NEs, where only Decision Trees technique was used within the hybrid system. This technique was applied to a limited set of selected features. The experimental results were promising and assure the quality of the prototype [1]. As a continuation, we extend the ML feature space to include morphological and contextual features. In addition to Decision Trees, we investigate two more ML algorithms: Support Vector Machines and Logistic Regression in the recognition of 11 different types of NEs [20]. In this paper, we report our experience with Arabic Person name recognition in particular. A wider standard datasets are used to evaluate our system. In [20], we reported a set of experimental results which was an indicative of a better system's performance in term of accuracy. Thereafter, more experiments and analysis of results are conducted to assess the quality of the hybrid system by means of standard evaluation metrics.

The structure of the remainder of this paper is as follows. Section 2 provides some background on NER, while Section 3 gives a literature review. Section 4 describes the method followed for data collection. Section 5 illustrates the architecture of the proposed system and then describes in details the main components. The experimental results are reported and discussed in Section 6. Section 7 concludes this paper and gives directions for future work.

2   Background

2.1   NER and NLP Applications

In the 1990s, at the Message Understanding Conferences (MUC), the task of NER was firstly introduced by the research community. Three main NER subtasks were defined at the 6th MUC: ENAMEX (i.e. Person, Location and Organization), TIMEX (i.e. temporal expressions), and NUMEX (i.e. numerical expressions).

The role of NER within NLP applications differs from an application to another. Examples of those NLP applications (but not limited to) are listed below:

- **Information Retrieval (IR).** IR is the task of identifying and retrieving relevant documents out of a database according to an input query [10]. There are two possible ways that IR can benefit from NER: 1) recognizing the NEs within the query, 2) recognizing the NEs within the documents to extract the relevant documents tak-

ing into account their classified NEs. For example, the word "واشنطن" waAšinTun[1] "Washington" can be recognized as a Location NE or a Person NE, hence the correct classification will lead to the extraction of the relevant documents.

- **Machine Translation (MT).** MT is the task of translating a text into another natural language. NEs need special handling in order to be translated correctly. Hence, the quality of NE translation would become an integral part that enhances the performance of the MT system [4]. In the translation from Arabic to Latin languages, Person names (NEs) can also be found as regular words (non-NEs) in the language without any distinguishing orthographic characteristics between the two surface forms. For example, the surface word "وفاء" wafaA' can be used in Arabic text as a noun which means trustfulness and loyalty, and also as a Person name.
- **Question Answering (QA).** QA application is closely related to IR but with more sophisticated results. A QA system takes questions as input and returns concise and precise answers. NER can be exploited in recognizing NEs within the questions to help identifying the relevant documents and then extracting the correct answers [16]. For instance, the words "إرنست ويونغ" Ăirnist wayuwnγ "Ernst & Young" may be classified as Organization or Person NEs according to the context.

2.2   Arabic Language Characteristics

The main characteristics of Arabic that pose non-trivial challenges for NER are:

- **No Capitalization:** Capitalization is not a feature of Arabic script, unlike Latin languages where NEs usually begins with capital letter. Therefore, the usage of the capitalization feature is not an option in Arabic NER. However, the English translation of Arabic words can be exploited as a feature indicator in this respect [13].
- **The Agglutinative Nature:** Arabic language has a high agglutinative nature in which a word may consist of prefixes, lemma and suffixes in different combination, which results in a very complicated morphology [2].
- **Optional Short Vowels:** In theory, short vowels, or diacritics, are needed for pronunciation and disambiguation. However, practically, most modern standard Arabic texts do not include diacritics, and therefore, a surface form of a word may refer to two or more different meanings according to the context they appear in.
- **Spelling Variants:** In Arabic script, the word may be spelled differently and still refers to the same word with the same meaning, creating a many-to-one ambiguity, e.g. the word "جرام" jrAm "Gram" can also be written as "غرام" γrAm.
- **Lack of Linguistic Resources:** There is a limitation in the number of Arabic linguistic resources (corpora (i.e. datasets) and gazetteers (i.e. predefined lists of NEs and keywords)) that are publicly available free for the research purposes. Many of the available corpora are neither annotated with NEs nor include sufficient number of NEs which make them unsuitable for NER task. Therefore, researchers tend to spend tangible efforts to annotate/acquire and verify their own Arabic linguistic resources in order to train and test their systems.

---

[1]   We used Habash-Soudi-Buckwalter transliteration scheme [15].

3   Literature Review of Arabic NER

In this section, we focus on the Arabic NER systems that have the capability to recognize Person names. They are divided to Rule-based and ML-based systems.

3.1   Rule-Based NER

Rule-based NER systems depend on local handcrafted linguistic rules to identify NEs within texts using linguistic and contextual clues, and indicators [24]. Such systems exploit gazetteers/dictionaries as auxiliary clues to the rules. The rules are usually implemented in the form of regular expressions or finite-state transducers [18].

[17] has presented TAGARAB system which is one of the early attempts to tackle Arabic NER. It is a rule-based system where a pattern matching engine is combined with a morphological tokenizer to identify Person, Organization, Location, Number and Time NEs. The empirical results show that combining the NE finder with the morphological tokenizer improves the performance of the system.

[18] has developed an Arabic component under NooJ linguistic environment to enable Arabic NER. The NE finder exploits a set of gazetteers and indicator lists to support rules construction. The system identifies NEs of types: Person, Location, Organization, Currency, and Temporal expressions. The system utilizes morphological information in the recognition of unclassified proper nouns as well.

Another work adopting the rule-based approach for NER is the one developed by [23] called PERA. It is a grammar-based system which is built for identifying Person names in Arabic scripts. PERA is composed of three components: gazetteers, local grammar and filtration mechanism. Whitelists of complete Person names are provided to extract the matching names regardless of the grammars. Afterwards, the input text is presented to the local grammar to identify the rest of Person NEs using the gazetteers. Finally, the filtration mechanism is applied on NEs detected through certain grammatical rules to exclude ambiguous and invalid NEs. PERA achieved satisfactory results when applied to the ACE and Treebank Arabic datasets.

As a continuation of [23] research work, NERA system was introduced in [24, 25]. NERA is a rule-based system that is capable of recognizing NEs of 10 different types: Person, Location, Organization, Date, Time, ISBN, Price, Measurement, Phone Numbers and Filenames. The system was implemented in the FAST ESP framework, where the system has three components as PERA [23] with the same functionalities. The Authors have constructed their own corpora from different resources in order to have a representative number of instances for each NE type.

[12] has proposed a rule-based NER system that integrates pattern matching with morphological analysis to extract Arabic Person names. The pattern matching engine utilizes lists of keywords without using predefined lists of Person names. The performance of the system was compared to PERA [23] despite the fact that PERA is evaluated using different datasets than the ones used for [12]'s system evaluation.

[26] has introduced a rule-based Arabic NER system to extract Person, Location and Organization NEs. The system is composed of three phases: morphological preprocessing, looking up known NEs and using local grammar to extract unknown NEs.

3.2   Machine Learning Based NER

ML-based NER systems take advantage of the ML algorithms in order to learn NE tagging decisions from annotated texts. The most common approach used in ML-based NER is Supervised Learning (SL) approach which represents the NER problem as a classification task. Among the most common SL techniques utilized for NER are Support Vector Machines (SVM), Conditional Random Fields (CRF), Maximum Entropy (ME), Hidden Markov Models (HMM) and Decision Trees [19].

[5] has developed an Arabic NER system, ANERsys 1.0, which uses ME. The authors have built their own linguistic resources which have become a de facto standard in Arabic NER literature: ANERcorp (i.e. an annotated corpus) and ANERgazet (Person, Location and Organization gazetteers). The features used by the system are lexical, contextual and gazetteers features. The system can recognize four types of NEs: Person, Location, Organization and Miscellaneous. The system raised some difficulties when detecting NEs that are composed of more than one token/word; hence [6] developed ANERsys 2.0, which employs a 2-step mechanism for NER: 1) detecting the start and the end points (boundaries) of each NE, and 2) identifying the NE type. [7] has applied CRF instead of ME as an attempt to improve the performance. The feature set used in ANERsys 2.0 was used in the CRF-based system. The features are POS tags and base phrase chunks (BPC), gazetteers and nationality. The CRF-based system achieves higher results in terms of accuracy. [8] has developed another NER system based on SVM. The features used are contextual, lexical, morphological, gazetteers, POS-tags and BPC, nationality and the corresponding English capitalization. The system has been evaluated using ACE Corpora and ANERcorp.

A simplified feature set has been proposed by [3] to be utilized in Arabic NER. They relied on CRF to recognize three types of NEs: Person, Location and Organization. The system considers only surface features without taking into account any other type of features. The system is evaluated using ANERcorp and ACE2005 dataset.

[9] investigated the sensitivity of different NE types to various types of features, i.e. in [8]. They build multiple classifiers for each NE type adopting SVM and CRF approaches. ACE datasets are used in the evaluation process. According to their findings, it cannot be stated whether CRF is better than SVM or the vice versa in Arabic NER. Each NE type is sensitive to different features and each feature plays a role in recognizing the NE in different degrees. Further studies, [10, 11], have confirmed as well the importance of considering language-independent and language-specific features in Arabic NER.

[2] integrated two ML approaches to handle Arabic NER including CRF and bootstrapping pattern recognition. The feature set used includes word-level features, POS tag, BPC, gazetteers and morphological features. The system is developed to extract 10 types of NEs: Person, Location, Organization, Job, Device, Car, Cell Phone, Currency, Date and Time. The system outperforms LingPipe recognizer when both are applied to ANERcorp dataset.

4   Data Collection

The linguistic resources are of two main categories: corpora and gazetteers. The corpora used in this research are Automatic Content Extraction[2] (ACE) corpora and ANERcorp[3] dataset. In the literature, they are commonly used for evaluation as well as comparison with existing systems. The dataset files have been prepared and transformed using our tag schema and in XML format. An example of a Person name in our tag schema is: <Person>هناء</Person>. The three ACE corpora used in this research are ACE 2003 (Newswire (NW) and Broadcast News (BN)) and ACE 2004 (NW) datasets. ANERcorp is an annotated dataset provided by [5]. In this study, the total number of annotated Person NEs covered by all datasets is 6,695 as demonstrated in Table 1. Another type of linguistic resources used is gazetteers. The gazetteers required for Person name recognition are collected as is from [24].

Table 1. The number of Person NEs in each reference dataset

Dataset NE type	ACE 2003 NW	ACE 2003 BN	ACE 2004 NW	ANERcorp	Total
Person	711	517	1865	3602	**6695**

5   The System Architecture

In this article, we propose a hybrid architecture that is demonstrably better than the rule-based or ML-based systems individually. Figure 1 illustrates the architecture of the proposed hybrid system for Arabic. The system consists of two sequential loosely coupled components: 1) a rule-based component that produces NE labels based on lists of NEs/keywords and contextual rules, and 2) an ML-based post-processor intended to make use of rule-based component's NE decisions as features aiming at enhancing the overall performance of the NER task.

5.1   The Rule-Based Component

The rule-based component is a reproduction of the NERA system [24] using the GATE framework[4]. It consists of three main modules: Whitelists (lists of full names), Grammar Rules and a Filtration mechanism (blacklists of invalid names) as illustrated in Figure 1. In GATE, the rule-based component works as a corpus pipeline where a corpus is processed through an Arabic tokenizer, resources including a list of gazetteers, and local grammatical Rules (implemented as finite-state transducers).

---

[2]   Available for us under license agreement from the Linguistic Data Consortium (LDC).
[3]   Available to download on `http://www1.ccls.columbia.edu/~ybenajiba/downloads.html`
[4]   GATE is freely available at the web link: `http://gate.ac.uk/`

Figure 2 illustrates an example of the Person name rules utilized by the rule-based component. The function of the rule in figure 2 is recognizing expressions that start with "ابو" or "ام" then followed by a First Person Name with the possibility of having a First, Middle or Last Name afterwards. Examples of Person names extracted by this rule: "ابو حسن" (The father of Hassan), and "ام عمر طه" (The mother of Omar Taha).
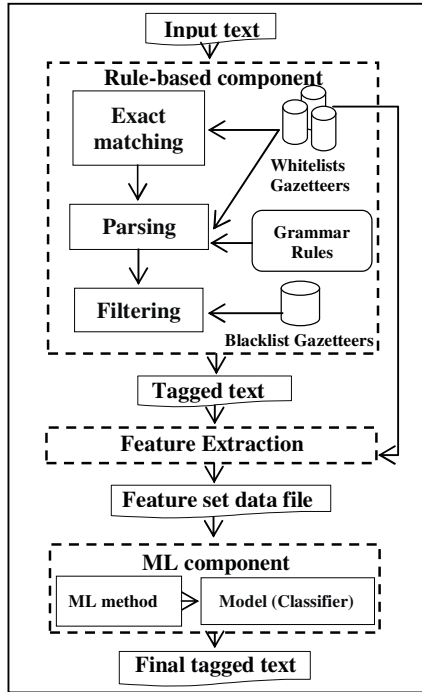

Fig. 1. The Overall Architecture of the Hybrid System

Person Rule in the form of regular expression:

((ابو|ام) + First Name + (First Name | Middle Name | Last Name)?)

Person Rule as implemented in GATE:

```
Rule: PersonRule5
Priority:14
(   ({Token.string == "ابو"}|{Token.string == "ام"})
{Lookup.majorType == "Firsts_v"}
({Lookup.majorType == "Firsts_v"}|{Lookup.majorType == "Middle_vv"}
 |{Lookup.majorType == "Lasts_v"})?    ) :Per
->
:Per.Person={rule="PersonRule1}, :Per.Person={rule="PersonRule5}
```

Fig. 2. An example of Person name rule within the rule-based component

5.2   The ML-Based Component

The ML-based component depends on two main aspects: feature engineering and selection of ML classifiers. The first aspect involves the selection and extraction of classification features. The features explored are divided into various categories: rule-based features (i.e. derived from the rule-based component's decisions), morphological features, POS features, Gazetteer features, contextual features, and word-level features. Exploring different types of features allow studying the effect of each feature category on the overall performance of the proposed system. The second aspect concerns the ML technique to be used in the training, testing and prediction phases. Three ML techniques (Decision Trees, SVM, and Logistic Regression) have been examined individually to reach a conclusion with regards to the best approach to work in our hybrid system. The first two techniques were chosen for their high performance in NER, while we decided to investigate the effect of the third technique on the proposed system's performance. WEKA[5] is utilized as the environment of the ML task. The classification features used by the ML-based component for Person name recognition are as follows:

— *Rule-based features:* The NE type predicted by the rule-based component for the targeted word as well as the NE types for the two immediate left and right neighbors of the candidate word, i.e. NE type for a sliding window of size 5.
— *Morphological Features:* a set of 13 features generated by MADA [14].
— *Word length flag:* A binary feature to indicate whether the word length ≥ 3.
— *Dot flag:* A binary feature to indicate whether the word has adjacent dot.
— *Capitalization flag:* A binary feature to indicate the existence of capitalization information on the English gloss (translation) corresponding to the Arabic word.
— *Check Gazetteers feature flags*: A binary feature to represent whether the word (or left/right neighbour of targeted word) belongs to the Gazetteer set.
— *POS tag:* part-of-speech tag of the targeted word estimated by MADA[6].
— *Nominal flag:* A binary feature to represent whether POS tag is a Noun/Proper Noun.
— Actual NE tag of the word: it is used along with other features for training the classification model. It is also used as a reference for calculating the accuracy.

6   Experimental Results

6.1   Experimental Setup

We conduct testing and evaluation experiments to test the rule-based component and compare it to the hybrid system. At the level of the hybrid system, experiments are subdivided at three dimensions: the corpora, the ML classifiers, and the

---

[5]   WEKA is available on `www.cs.waikato.ac.nz/ml/weka/`
[6]   MADA is available on: `http://www1.ccls.columbia.edu/MADA/`

inclusion/exclusion of feature groups. The reference datasets are the initial datasets described with their tagging details in Section 4 including ACE corpora and ANER-corp.

The performance of the rule-based component is evaluated using GATE built-in evaluation tool, so-called *AnnotationDiff*. On the other hand, the ML-based component uses three different functions (or classifiers) to be applied to the datasets, including Decision trees, SVM and Logistic regression approaches which are available in WEKA via J48, LibSVM and Logistic classifiers respectively. In this research, 10-fold cross validation is chosen to avoid overfitting. The WEKA tool provides the functionality of applying the conventional k-fold cross-validation for evaluation.

6.2   Experiments and Results

A number of experiments have been conducted to evaluate the performance of the proposed system when applied to different datasets. We group similar features together according to the nature of the feature type. We examined six settings of feature groups in order to study their effect on the overall performance: when all features are considered, and when all-but-one feature group are considered. They are:

1. All Features: all features are considered.
2. W/O RB: excluding the rule-based (RB) features (i.e. pure ML-based mode).
3. W/O MF: excluding the morphological features.
4. W/O POS: excluding POS feature.
5. W/O GAZ: excluding Gazetteers features.
6. W/O NbG: excluding neighbors' related features within the Gazetteers features.

The baseline in all experiments is the performance of the pure rule-based component.

Table 2 shows the system's performance in terms of F-measure when applied on ACE2003 (NW & BN), ACE2004 NW, and ANERcorp datasets in order to extract Person NEs. According to the empirical results illustrated in this table, the highest performance of our system when applied on ACE2003 NW and ANERcorp datasets are achieved by J48 classifier when the 6th feature setting is used (i.e. without neighboring features), while using J48 classifier with the 1st setting (i.e. all Features are used) leads to the highest performance when applied on ACE2003 BN and ACE2004 NW datasets.

The experimental results show that the adaptation of the hybrid approach leads to the highest performance. Also, the decision trees function has proved its comparatively higher efficiency as a classifier in our hybrid system. In comparison, the results achieved by [5], [6], [7] and [1] when applied on ANERcorp, have shown that our system performs demonstrably better as illustrated by Table 3. As it can be noticed, our hybrid system outperforms the other systems in extracting Person NEs from ANERcorp dataset. It is worth noting that a comparison between our results and [8, 9]'s results is not possible because their published evaluation lacks sufficient details.

Table 2. The results of applying the proposed hybrid system on ACE2003 (NW & BN), ACE2004 (NW), & ANERcorp datasets in order to extract Person names

		ACE2003 NW	ACE2003 BN	ACE2004 NW	ANERcorp
		F-measure	F-measure	F-measure	F-measure
Rule-based (baseline)		0.7548	0.7646	0.3455	0.6965
J48	All Features	0.932	0.903	0.824	0.944
	W/O RB	0.913	0.886	0.817	0.921
	W/O MF	0.93	0.917	0.812	0.941
	W/O POS	0.924	0.91	0.776	0.94
	W/O GAZ	0.906	0.909	0.785	0.928
	W/O NbG	0.934	0.902	0.82	0.945
Libsvm	All Features	0.919	0.898	0.804	0.942
	W/O RB	0.869	0.844	0.793	0.912
	W/O MF	0.928	0.902	0.805	0.939
	W/O POS	0.919	0.891	0.758	0.939
	W/O GAZ	0.888	0.883	0.753	0.926
	W/O NbG	0.921	0.895	0.795	0.943
Logistic	All Features	0.912	0.902	0.806	0.943
	W/O RB	0.87	0.846	0.799	0.917
	W/O MF	0.903	0.896	0.795	0.935
	W/O POS	0.904	0.896	0.766	0.925
	W/O GAZ	0.889	0.896	0.774	0.919
	W/O NbG	0.906	0.9	0.8	0.937

Table 3. The results of ANERsys 1.0, ANERsys 2.0, CRF-based system [7] and Abdallah et al. [1]'s system compared to our hybrid system's highest performance when applied to ANERcorp dataset in order to extract Person names

System	Person		
	Precision	Recall	F-measure
ANERsys 1.0 [5]	0.5421	0.4101	0.4669
ANERsys 2.0 [6]	0.5627	0.4856	0.5213
CRF-based system [7]	0.8041	0.6742	0.7335
Abdallah et al. [1]	0.949	0.9078	0.928
Our Hybrid System (J48)	**0.949**	**0.942**	**0.945**

7   Conclusion and Future Work

In the literature, the use of either rule-based approach or pure ML-based approach is considered a successful approach for Arabic NER in general and Arabic Person name

recognition in particular. Our proposed hybrid approach is distinct from these approaches in that the ML-based subsystem can make use of rule-based decisions determined by the rule-based subsystem in order to improve the performance of Arabic Person name recognition. A number of extensive experiments are conducted on three different dimensions including the dataset, the feature set, and the ML technique used to evaluate the performance of our domain-independent system when applied on a variety of standard datasets. The experimental results prove that the hybrid approach outperforms the pure Rule-based approach and the pure ML-based approach. Moreover, the proposed system outperforms the state-of-the-art of the Arabic Person NER when applied to ANERcorp standard dataset with Precision of 0.949, Recall of 0.942 and F-measure of 0.945 for Person NEs.

In future work, we intend to enhance the gazetteers and explore the possibility of improving the system by adding more lists of predefined Person NEs. There is also a space for improving the local grammar rules implemented within the rule-based component through analyzing the hybrid system's output in a way to automate the enhancement process. We are also considering the possibility of investigating other different ML techniques with our hybrid system.

References

1. Abdallah, S., Shaalan, K., Shoaib, M.: Integrating Rule-Based System with Classification for Arabic Named Entity Recognition. In: Gelbukh, A. (ed.) CICLing 2012, Part I. LNCS, vol. 7181, pp. 311–322. Springer, Heidelberg (2012)
2. AbdelRahman, S., Elarnaoty, M., Magdy, M., Fahmy, A.: Integrated Machine Learning Techniques for Arabic Named Entity Recognition. IJCSI 7, 27–36 (2010)
3. Abdul-Hamid, A., Darwish, K.: Simplified Feature Set for Arabic Named Entity Recognition. In: Proceedings of the 2010 Named Entities Workshop, pp. 110–115 (2010)
4. Babych, B., Hartley, A.: Improving Machine Translation Quality with Automatic Named Entity Recognition. In: Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT (EAMT 2003), pp. 1–8 (2003)
5. Benajiba, Y., Rosso, P., BenedíRuiz, J.M.: ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp. 143–153. Springer, Heidelberg (2007)
6. Benajiba, Y., Rosso, P.: ANERsys 2.0: Conquering the NER task for the Arabic language by combining the Maximum Entropy with POS-tag information. In: Proceedings of Workshop on Natural Language-Independent Engineering, IICAI 2007, pp. 1814–1823 (2007)
7. Benajiba, Y., Rosso, P.: Arabic Named Entity Recognition using Conditional Random Fields. In: Proceedings of LREC 2008 (2008)
8. Benajiba, Y., Diab, M., Rosso, P.: Arabic Named Entity Recognition: An SVM-Based Approach. In: Proceedings of (ACIT 2008), pp. 16–18 (2008)

9. Benajiba, Y., Diab, M., Rosso, P.: Arabic Named Entity Recognition Using Optimized Feature Sets. In: Proceedings of EMNLP 2008, pp. 284–293 (2008)
10. Benajiba, Y., Diab, M., Rosso, P.: Arabic Named Entity Recognition: A Feature-Driven Study. IEEE Transactions on Audio, Speech and Language Processing 17, 926–934 (2009)
11. Benajiba, Y., Diab, M., Rosso, P.: Using Language Independent and Language Specific Features to Enhance Arabic Named Entity Recognition. The International Arab Journal of Information Technology 6, 464–473 (2009)
12. Elsebai, A., Meziane, F., BelKredim, F.Z.: A Rule Based Persons Names Arabic Extraction System. In: Communications of the IBIMA, pp. 53–59 (2009)
13. Farber, B., Freitag, D., Habash, N., Rambow, O.: Improving NER in Arabic Using a Morphological Tagger. In: Proceedings of Workshop on HLT & NLP within the Arabic World (LREC 2008), pp. 2509–2514 (2008)
14. Habash, N., Owen, R., Ryan, R.: MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. In: Proceedings of the 2nd International Conference on Arabic Language Resources and Tools, MEDAR (2009)
15. Habash, N., Soudi, A., Buckwalter, T.: On Arabic Transliteration. In: Arabic Computational Morphology: Knowledge-based and Empirical Methods, pp. 15–22 (2007)
16. Hamadene, A., Shaheen, M., Badawy, O.: ARQA: An Intelligent Arabic Question Answering System. In: Proceedings of ALTIC 2011 (2011)
17. Maloney, J., Niv, M.: TAGARAB: A Fast, Accurate Arabic Name Recognizer Using High-Precision Morphological Analysis. In: Proceedings of the Workshop on Computational Approaches to Semitic Languages (Semitic 1998), pp. 8–15 (1998)
18. Mesfar, S.: Named Entity Recognition for Arabic Using Syntactic Grammars. In: Kedad, Z., Lammari, N., Métais, E., Meziane, F., Rezgui, Y. (eds.) NLDB 2007. LNCS, vol. 4592, pp. 305–316. Springer, Heidelberg (2007)
19. Nadeau, D., Sekine, S.: A Survey of Named Entity Recognition and Classification. Lingvisticae Investigationes 30, 3–26 (2007)
20. Oudah, M.M., Shaalan, K.: A Pipeline Arabic Named Entity Recognition Using a Hybrid Approach. In: Proceedings of COLING 2012, pp. 2159–2176 (2012)
21. Petasis, G., Vichot, F., Wolinski, F., Paliouras, G., Karkaletsis, V., Spyropoulos, C.D.: Using Machine Learning to Maintain Rule-based Named-Entity Recognition and Classification Systems. In: Proceeding of Association for Computational Linguistics, pp. 426–433 (2001)
22. Shaalan, K.: Rule-based Approach in Arabic Natural Language Processing. IJICT 3, 11–19 (2010)
23. Shaalan, K., Raza, H.: Person Name Entity Recognition for Arabic. In: Proceedings of the 5th Workshop on Important Unresolved Matters, pp. 17–24 (2007)
24. Shaalan, K., Raza, H.: Arabic Named Entity Recognition from Diverse Text Types. In: Nordström, B., Ranta, A. (eds.) GoTAL 2008. LNCS (LNAI), vol. 5221, pp. 440–451. Springer, Heidelberg (2008)
25. Shaalan, K., Raza, H.: NERA: Named Entity Recognition for Arabic. Journal of the American Society for Information Science and Technology 60, 1652–1663 (2009)
26. Zaghouani, W.: RENAR: A Rule-Based Arabic Named Entity Recognition System. ACM Transactions on Asian Language Information Processing 11, 1–13 (2012)

A Broadly Applicable and Flexible Conceptual Metagrammar as a Basic Tool for Developing a Multilingual Semantic Web

Vladimir A. Fomichov

Department of Innovations and Business in the Sphere of Informational Technologies,
Faculty of Business Informatics, National Research University Higher School of Economics,
Kirpichnaya str. 33, 105187 Moscow, Russia
`vfomichov@hse.ru, vfomichov@gmail.com`

Abstract. The paper formulates the problem of constructing a broadly applicable and flexible Conceptual Metagrammar (CM). It is to be a collection of the rules enabling us to construct step by step a semantic representation (or text meaning representation) of practically arbitrary sentence or discourse pertaining to mass spheres of human's professional activity. The opinion is grounded that the first version of broadly applicable and flexible CM is already available in the scientific literature. It is conjectured that the definition of the class of SK-languages (standard knowledge languages) provided by the theory of K-representations (knowledge representations) can be interpreted as the first version of broadly applicable and flexible CM. The current version of the latter theory is stated in the author's monograph published by Springer in 2010. The final part of the paper describes the connections with the related approaches, in particular, with the studies on developing a Multilingual Semantic Web.

Keywords: natural language processing, conceptual metagrammar, semantic markup language, algorithm of semantic-syntactic analysis, theory of K-representations, SK-languages, semantic representation, text   meaning representation, Multilingual Semantic Web, bioinformatics.

1   Introduction

During last decade, one has been able to observe in different parts of the world the permanent growth of interest in designing natural language (NL) interfaces to applied intelligent systems and in constructing other kinds of NL processing systems, or linguistic processors. In particular, a number of projects being useful for practice are described in [1-7].

One of the most acute and large-scale problems is to endow the existing Web with the ability of extracting information from numerous sources in various natural languages (of cross-language information retrieval) and of constructing NL-interfaces to a number of knowledge repositories recently developed under the framework of the Semantic Web project [2, 8-12].

The aim of this paper is to introduce the notion of a broadly applicable and flexible Conceptual Metagrammar (CM) and to ground the opinion that the first version of such CM does already exist. More exactly, that the definition of the class of SK-languages (standard knowledge languages) provided by the theory of K-representations (knowledge representations) [9-13] can be interpreted as the first version of a broadly applicable and flexible CM. The final part of the paper discusses the connections with the related approaches.

2   Problem Statement

The analysis of many relatively recent publications on semantic processing of NL-texts by computer intelligent systems evokes the astonishment concerning a huge gap between the scale of the problems to be solved and the used formal means for reflecting semantics of NL-texts.

For instance, Harrington and Clark [6] describe the ASKNet system designed in the Oxford University Computing Laboratory. This system is able to automatically extract semantic information from the texts in English and, as a result of integrating this information, constructs a large-scale semantic network (SN). For this, the ASKNet system uses a number of existing NL processing tools and an enriched spreading activation theory. The system is able to create SN consisting of over 1.5 million nodes and 3.5 million edges in less than three days.

According to [6], the underlying semantic theory is Discourse Representation Theory (DRT). But DRT is a syntactic version of first-order logic (FOL), and numerous restrictions of FOL are well known (see, e.g., [11]). Due to these restrictions, the authors of the discussed paper are forced to draw a picture (a kind of a semantic net) for representing semantic structure of the discourse T1 = "Yesterday John heard that ABC Inc. hired Susan. Bob decided that ABC Inc. will move to London. Susan met Bob twice".

The reason for drawing a picture for this simple text is that FOL allows for constructing the simplest formulas only of the form $P(t_1, ..., t_n)$, where n is not less than 1, P is the name of an n-ary predicate, $t_1, ..., t_n$ are terms (so no one element from $t_1, ..., t_n$ can be a formula). That is why FOL, and, as consequence, DRT, don't provide adequate formal means of describing semantic structure of the sentences with direct or indirect speech, etc.

Due to the lack (subjectively perceived) of efficient formal tools for reflecting many semantic expressive mechanism of NL, the scholars usually consider in their papers only very small sublanguages of NL, avoiding the consideration, in particular, of (a) compound designations of sets, sequences, and concepts, (b) the infinitive or gerundial constructions expressing the goals, commitments, wishes, commands; (c) the discourses with references to the meaning of preceding phrases or larger parts of a discourse, etc.

The analysis of the scientific literature of the design of semantics-oriented NLPSs and a Multilingual Semantic Web provides serious arguments in favour of putting forward the following conjecture: *it is high time for creating a new paradigm* for

considering numerous theoretical problems encountered while constructing and processing various conceptual structures associated with Web-based informational sources: semantic representations of written and spoken texts' fragments (in other terms, text meaning representations); high-level conceptual descriptions of visual images; knowledge pieces stored in ontologies; the content of messages sent by computer intelligent agents, etc.

What can be a key to solving this problem? We do know that, using NL, we are able to describe various pieces of knowledge, the content of a visual images, the content of a film, etc. That is why it can be conjectured that a key to elaborating a new paradigm of the described kind could be the construction of a broadly applicable and flexible Conceptual Metagrammar (CM). It is to be a collection of the rules enabling us to construct step by step a semantic representation (or text meaning representation) of practically arbitrary sentence or discourse pertaining to mass spheres of professional activity of people.

The prefix "meta" in the term "metagrammar" means that such rules are to use the information associated with the classes of conceptual items. That is why we should be able to employ the same system of rules with different conceptual vocabularies.

3   Methodology

3.1   Shortly about the Theory of K-Representations

As far as in the middle of the 1960s, the researchers had practically the only formal approach to describing structured meanings (SMs) of NL-texts : the first-order logic (FOL). Due to numerous restrictions of FOL, the search for more powerful and flexible formal means for describing SMs of NL-texts was started in the second half of the 1960s. As a result, a number of new theories have been developed, first of all, the Theory of Generalized Quantifiers (TGQ), Discourse Representation Theory (DRT), Theory of Semantic Nets (TSN), Theory of Conceptual Graphs (TCG), Episodic Logic (EL), and Theory of K-representations (knowledge representations). The latter theory is an original theory of designing semantic-syntactic analysers of NL-texts with the broad use of formal means for representing input, intermediary, and output data [9-13]. This theory also contributes to the development of logic-informational foundations of (a) Semantic Web of a new generation, (b) E-commerce, and (c) multi-agent systems theory (agent communication languages) [11-12].

In order to understand the principal distinction of the theory of K-representations from other mentioned approaches to formalizing semantics of NL, let's consider an analogy. Bionics studies the peculiarities of the structure and functioning of the living beings in order to discover the new ways of solving certain technical problems. Such theories as TGQ, DRT, TSN, TCG, EL and several other theories were elaborated on the way of expanding the expressive mechanisms of FOL. To the contrary, the theory of K-representations was developed as a consequence of analysing the basic expressive mechanisms of NL and putting forward a conjecture about a system of partial operations on conceptual structures underpinning these expressive

mechanisms. Of course, the idea was to develop a formal model of this system being compatible with FOL.

The *first basic constituent* of the theory of K-representations is the theory of SK-languages (standard knowledge languages). The kernel of this theory is a mathematical model describing a system of such 10 partial operations on structured meanings (SMs) of natural language texts (NL-texts) that, using primitive conceptual items as "blocks", we are able to build SMs of arbitrary NL-texts (including articles, textbooks, etc.) and arbitrary pieces of knowledge about the world. The analysis of the scientific literature shows that today the class of SK-languages opens the broadest prospects for representing SMs of NL-texts in a formal way.

The *second basic constituent* of the theory of K-representations is a broadly applicable mathematical model of a linguistic database [9, 11]. The *third basic constituent* of the theory of K-representations is several complex, strongly structured algorithms carrying out semantic-syntactic analysis of texts from some practically interesting sublanguages of NL. The algorithm *SemSynt1* transforms a NL-text in its semantic representation being a K-representation [11]. The input texts (statements, commands, and questions of many kinds) can be from the English, German, and Russian languages. This algorithm is implemented by means of a program in the language PYTHON.

The paper [14] describes an application of the theory of K-representations to the elaboration of a new approach to semantic search of documents on the Web. The subject of the paper is semantic processing of the requests about the achievements or failures of the organizations (firms, etc.) and people. A generalized request of the end user is transformed into a set of concrete requests, it is done with the help of a goals base storing the semantic representations of the goals of active systems. A model of a goals base is constructed with the help of the theory of K-representations.

3.2   Formalization of Basic Assumptions about Primary Items of Conceptual Level

The first part of the theory of SK-languages is a mathematical model describing a system of primary conceptual units used by an applied intelligent system, in particular, by a NL processing system. This model defines (with the help of a rather long sequence of auxiliary steps) a new class of formal objects called *conceptual bases (c.b.),* where each concrete c.b. is constructed for a certain group of application domains. Each c.b. *B* is equivalent to a system of the form $(c_1, \ldots, c_{15})$ with the components $c_1, \ldots, c_{15}$ being mainly finite or countable sets of symbols and distinguished elements of such sets. In particular, $c_1 = St$ is a finite set of symbols called *sorts* and designating the most general considered notions (concepts); $c_5 = X = X(B)$ is a countable set of strings used as elementary blocks for building knowledge modules and semantic representations (SRs) of texts (see the examples below); X is called a primary informational universe; $c_6 = V$ is a countable set of variables; $c_8 = F$ is a subset of X whose elements are called functional symbols.

3.3   The Essence of a Model of a System Consisting of Ten Partial Operations on Conceptual Structures

Each c.b. *B* determines three classes of formulas, the first class *Ls(B)* being considered as the principal one and being called *the SK-language (standard knowledge language) in the basis B*. Its strings (called K-strings) are convenient for building SRs of NL-texts. We'll consider below only the formulas from the first class *Ls(B)*. If *Expr* is an expression in natural language and a K-string *Semrepr* can be interpreted as a semantic representation of *Expr*, then *Semrepr* will be called a K-representation (KR) of the expression *Expr*.

For determining for arbitrary c.b. *B* three classes of formulas, a collection of inference rules P[0], P[1], ... , P[10] is defined. The rule P[0] provides an initial stock of formulas from the first class. For arbitrary c.b. *B*, let *Degr(B)* be the union of all Cartesian m-degrees of *Ls(B),* where *m* is not less than 1. Then the meaning of the rules of constructing well-formed formulas P[1], ..., P[10] can be explained as follows: for each k from 1 to 10, the rule P[k] determines a partial unary operation *Op[k]* on the set *Degr(B)* with the value being an element of *Ls(B)*.

Example. There is a conceptual basis B possessing the following properties. The primary informational universe $X = X(B)$ includes the conceptual items *prophase, prometaphase, metaphase, nanaphase, telophase* describing five distinct stages of mitosis (the process of somatic cell division, during which the nucleus also divides) and the conceptual items *China, India, Sri_Lanka*. Hence the value of the partial operation *Op[7]* (it governs the use of logical connectives ∧ - AND and ∨ - OR) on the six-tuple < ∧, *prophase, prometaphase, metaphase, nanaphase, telophase* > is the string *Semexpr1* of the form *(prophase∧ prometaphase ∧ metaphase ∧ nanaphase ∧ telophase),* and the value on the four-tuple < ∨, *China, India, Sri-Lanka* > is the K-string *(China ∨ India ∨ Sri-Lanka).*

Let *X(B)* also include the item *mitosis* and the designation of a binary relation *Stages-relation*. Then the K-string *Stages-relation (mitosis, Semexpr1)* is the result of applying the partial operation P[4] to the operands *Stages-relation, mitosis, and Semexpr1*. Besides, let *X(B)* include the items *article1* (a paper), *article2* (a manufactured article), and *h1 = article2, h2 = Kind1(certn article2, ceramics), h3 = (Country1(certn article2) = (China ∨ India ∨ Sri-Lanka)), h4 = article2 * (Kind1, ceramics) (Country1, (China ∨ India ∨ Sri_Lanka))* are the elements of *Ls(B)*. Then the K-string *h4* is the result of applying the partial operation P[8] to the operands *h1, h2, h3*.

Ls(B) includes the string *h5* of the form *certn h4*, being the result of applying the operation P[1] to the operands *certn* and *h4*. The item *certn* denotes the meaning of the expression "a certain", and the string *h5* is interpreted as a designation of a manufactured article being a kind of ceramics and produced in China, India, or Sri-Lanka.

Let h6 be the string of the form *(Height(h5) =14/cm).* Then *h6* belongs to *Ls(B)* and is the result of applying the partial operation P[3] to the operands *Height(h5)* and *14/cm.* Thus, the essence of the basic model of the theory of SK-languages is as follows: this model determines a partial algebra of the form (*Degr(B), Operations(B)*) , where

Degr(B) is the carrier of the partial algebra, *Operations(B)* is the set consisting of the partial unary operations *Op[1], …, Op[10]* on *Degr(B)*.

The volume of the complete description in [11] of the mathematical model introducing, in essence, the operations *Op[1], …, Op[10]* on *Degr(B)* and, as a consequence, determining the class of SK-languages considerably exceeds the volume of this paper. That is why, due to objective reasons, this model can't be included in this paper. A short outline of the model can be found in [10].

4   Results

Let's consider the principal new expressive mechanisms introduced by the definition of the class of SK-languages.

4.1   Building Semantic Representations of Complex Discourses

During several last years, the significance of NL processing (NLP) technologies for informatics dealing with the problems of biology and medicine has been broadly recognized. As a consequence, the term BioNLP interpreted as the abbreviation for Natural Language Processing in Biology and Medicine was born. The formalization of NL semantics is a very acute problem of BioNLP. The attention of many researchers in this field is now attracted by the phenomena of the semantics of sentences and discourses [5]. That is why let's illustrate the new expressive possibilities provided by SK-languages on the example of building a semantic representation of a rather complex discourse pertaining to genetics.

Example. It is known that each individual possesses two genes being responsible for a particular characteristic (e.g., the height) in case of almost all characteristics (or traits). The genes responsible for the contrasting values of a characteristic (for instance, the values "tall" and "short" for the trait "height") are referred to as *allelomorphs*, or *alleles* for short. Some genes have more than two allelic forms, i.e. multiple alleles. In the case of the ABO blood group system, there are at least four alleles (A_1, A_2, B and O). An individual can possess any two of these alleles, which can be the same or different (AO, A_2B, OO, and so on).

With respect to this context, let's consider the discourse D1 = "Alleles are carried on homological chromosomes and therefore a person transmits only one allele for a certain trait to any particular offspring. For example, if a person has the genotype AB, he will transmit to any particular offspring either the A allele or the B allele, but never both or neither" [16].

Let S1 = "Alleles are carried on homological chromosomes", S2 = "therefore a person transmits only one allele for a certain trait to any particular offspring.", S3 = S1 and S2, S4 = "For example, if a person has the genotype AB, he will transmit to any particular offspring either the A allele or the B allele, but never both or neither". First of all, we'll construct a possible K-representation (KR) of the sentence S1 as the following string *Semrepr1*:

*(Entails((Alleles-relation (certn gene * (Part, certn person : y1) : x1, certn gene * (Part, y1) : x2) ∧ Location(x1, x3) ∧ Location(x2, x4) ∧ Semantic-descr((x3 ∧ x4),*

*chromosome * (Part, y1))), Homologous(x3, x4)) : P1 ∧ Correspondent-situation(P1, e1)).*

The K-string *Semrepr1* illustrates the following new properties of the theory of SK-languages: the possibilities (a) to construct the compound designations of the notions and of the objects qualified by these notions, (b) to use the logical connective ∧ (AND) for joining not only the semantic representations of the statements but also the designations of the objects, as in case of the substring *(x3 ∧ x4)*, (c) to associate the mark of a situation with the mark of the meaning of sentence describing this situation, as in case of the substring *Correspondent-situation(P1, e1)*.

A possible KR of the sentence S2 may be built as the string *Semrepr2* of the form

*(Cause(e1, e2) ∧ Corresponding-situation(P2, e2) ∧ (P2 = ∀ y2(person) ∀ y3(person * (Offspring-rel, y2)) ∀ x5(trait1 * (Possessed-by, y2)) (∃ x6 (gene * (Element, Alleles-function(x5))) Situation(e3, transmission1 * (Source1, y2)(Recipient1, y3)(Object-transmitted, x6)) ∧ ¬ ∃ x7 (gene * (Element, Alleles-function(x5))) (Situation(e4, transmission1 * (Source1, y2)(Recipient1, y3)(Object-transmitted, x7)) ∧ ¬ (x7 = x6))))).*

The symbols ∀ and ∃ in the K-string *Semrepr2* are the universal quantifier and the existential quantifier. We can see here that SK-languages allow for restricting the domain of a logical quantifier with the help of the expressions like *(person * (Offspring-rel, y2)), (trait1 * (Possessed-by, y2)), (gene * (Element, Alleles-function(x5)))*, and so on. At this point of our analysis we have the appropriate building blocks *Semrepr1* and *Semrepr2* for constructing a possible KR of the sentence S3 as the string *Semrepr3* of the form *(Semrepr1∧ Semrepr2) : P3*.

Now let's build a K-representation of the final sentence S4 in the context of the sentence S3. We see that the word combination "For example" from S4 encodes the reference to the meaning of the sentence S3. The system of ten partial operations on conceptual structures proposed by the theory of K-representations contains the operation Op[5] to be used just in such cases. This operation allows for constructing the formulas of the kind *form : var*, where the first operand *form* is a semantic description of an object (in particular, a SR of a statement), and *var* is a variable.

This operation was used for constructing the subformulas *certn gene * (Part, certn person : y1) : x1* and *certn gene * (Part, y1) : x2* of the formula *Semrepr1*; besides, for building the formula *Semrepr 3* from the operands *(Semrepr1∧ Semrepr2)* and *P3*. Now we can use the variable *P3* as a mark of the meaning of the sentence S3 in the following K-representation *Semrepr4* of the sentence S4:

*Example(P3, Entails(Situation(e4, posessing1 * (Owner1, arbitr person : y4)(Object1, certn genotype * (Designation, 'AB') : x7)), (Situation(e5, transmission1 * (Source1, y4)(Recipient1, arbitr person * (Offspring, y4) : y5)(Object-transmitted, (certn allele * (Designation, 'A") : x8 ∨ certn allele * (Designation, 'B") : x9))) ∧ Situation(e6, ¬ transmission1 * (Source1, y4)(Recipient1, y5)(Object-transmitted, (x8 ∧ x9))) ∧ Situation(e7, ¬ transmission1 * (Source1, y4)(Recipient1, y5) (Object-transmitted, NIL))))).*

Here *NIL* is the constant reflecting the meaning of the word "nothing".

Actually, we build a K-representation of the discourse D1 as a string of the form *((Semrepr1 ∧ Semrepr2) : P3 ∧ Semrepr4).*

To sum up, SK-languages allow for describing semantic structure of the sentences with direct and indirect speech and of the discourses with the references to the meanings of phrases and larger parts of a discourse, for constructing compound designations of the notions, sets, and sequences.

4.2   K-Representations of Complex Definitions of Notions

The analysis shows that the SK-languages possess a number of interrelated expressive mechanisms making them a convenient formal tool for building arbitrarily complex definitions of notions.

Example. Let Def1 = "A flock is a large number of birds or mammals (e.g. sheep or goats), usually gathered together for a definite purpose, such as feeding, migration, or defence". Def1 may have the K-representation *Expr1* of the form

> *Definition1 (flock, dynamic-group * (Qualitative-composition, (bird ∨ mammal ** *(Examples, (sheep ∧ goal)))), S1, (Estimation1(Quantity(S1), high) ∧ Goal-of-* *forming (S1, certain purpose * (Examples, (feeding ∨ migration ∨ defence))))).*

The analysis of this formula enables us to conclude that it is convenient to use for constructing semantic representations (SRs) of NL-texts: (1) the designation of a 5-ary relationship *Definition1*, (2) compound designations of concepts (in this example the expressions *mammal * (Examples, (sheep ∧ goal))* and *dynamic-group * (Qualitative-composition, (bird ∨ mammal * (Examples, (sheep ∧ goal))))* were used), (3) the names of functions with the arguments and/or values being sets (in the example, the name of an unary function *Quantity* was used, its value is the quantity of elements in the set being an argument of this function), (4) compound designations of intentions, goals; in this example it is the expression *certain purpose * (Examples, (feeding ∨ migration ∨ defence))* .

4.3   Object-Oriented K-Representations of Discourses

Example. Let Disc1 = "Yesterday John heard that ABC Inc. hired Susan. Bob decided that ABC Inc. will move to London. Susan met Bob twice" (this discourse is considered in [6]). It is possible to construct an object-oriented KR of Disc1 in the form

> *certain inf-object * (Kind1, text)(Authors, (B.Harrington ∧ S. Clark))(Mentioned-* *entities, (x1∧ x2 ∧ x3 ∧ x4 ∧ e1 ∧ e2 ∧ e3 ∧ e4 ∧ e5 ∧ e6))(Content-* *description,(Situation (e1, hearing1 * (Agent1, certain person * (Gender,* *male)(Name, "John") : x1)(Time, Yesterday(#current-moment#))(Content1, Situation* *(e2, hiring1 * (Agent2, certain firm1 * (Called, "ABC Inc.") : x2)(Object-person,* *certain person * (Gender, female)(Name, "Susan") : x3)))) ∧ Situation (e3, taking-* *conclusion * (Agent1, certain person * (Gender, male)(Name, "Bob") : x4))(Time,* *certain moment * (Before, #current-moment#) : t1) (Content1, Situation (e4, moving1* ** (Agent2, x2)(Destination, certain city * (Called, "London") : x5)(Time, certain* *moment * (Later, #current-moment#) : t2)))) ∧ Situation (e5, meeting1 *(Agent1,* *x3)(Participant2, x4)(Time, certain moment * (Before, current-moment#) : t3)) ∧*

*Situations-difference (e5, e6, (Time(e6, certain moment * (Before, current-moment#) : t4) ∧ Different (t3, t4))))) : inf001,*

where *inf001* is the unique mark of this information piece.

4.4   A Strategy of Developing a Multilingual Semantic Web

The process of endowing the existing Web with the ability of understanding many natural languages is an objective ongoing process. The analysis has shown that there is a way to increase the total successfulness, effectiveness of this global decentralized process. It would be especially important with respect to the need of cross-language conceptual information retrieval and question - answering. The proposed way is a possible new paradigm for the mainly decentralized process of endowing the existing Web with the ability of processing many natural languages.

The principal idea of a new paradigm is as follows. There is *a common thing* for the various texts in different natural languages. This common thing is the fact that *the NL-texts have the meanings.* The meanings are associated not only with NL-texts but also with the visual images (stored in multimedia databases) and with the pieces of knowledge from the ontologies.

That is why the great advantages are promised by the realization of the situation when a unified formal environment is being used in different projects throughout the world for reflecting structured meanings of the texts in various natural languages, for representing knowledge about application domains, for constructing semantic annotations of informational sources and for building high-level conceptual descriptions of visual images.

The analysis of the expressive power of SK-languages (see the chapters 3 – 6 of [11] shows that the SK-languages can be used as a unified formal environment of the kind. This idea underlies an original strategy of transforming step by step the existing Web into a Semantic Web of a new generation, where its principal distinguished feature would be the well-developed ability of NL processing; it can be also qualified as a Multilingual Semantic Web. The versions of this strategy are published in [11-12].

5   Discussion

The advantages of the theory of K-representations in comparison with first-order logic, Discourse Representation Theory, and Episodic Logic are, in particular, the possibilities: (1) to distinguish in a formal way objects (physical things, events, etc.) and notions qualifying them; (2) to build compound representations of notions; (3) to distinguish in a formal manner objects and sets of objects, concepts and sets of concepts; (4) to build complex representations of sets, sets of sets, etc.; (5) to describe set-theoretical relationships; (6) to effectively describe structured meanings (SMs) of discourses with references to the meanings of phrases and larger parts of discourses; (7) to describe SMs of sentences with the words "concept", "notion"; (8) to describe SMs of sentences where the logical connective "and" or "or" joins not the expressions-assertions but designations of things, sets, or concepts; (9) to build complex designations of objects and sets; (10) to consider non-traditional functions with arguments or/and values being sets of objects, of concepts, of texts' semantic

representations, etc.; (11) to construct formal analogues of the meanings of infinitives with dependent words and, as a consequence, to represent proposals, goals, commitments; (12) to build object-oriented representations of information pieces.

The items (3) - (8), (10) – (12) in the list above indicate the principal advantages of the theory of K-representations in comparison with the Theory of Conceptual Graphs. The global advantage of the theory of K-representations is that it puts forward a hypothesis about a system of partial operations on conceptual structures being sufficient and convenient for constructing semantic representations of sentences and discourses in NL pertaining to arbitrary fields of human's professional activity.

Taking into account the advantages listed above and the content of the considered examples, it is possible to conjecture that the theory of K-representations can be used as an adequate methodological basis for developing a new version of the system ASKNet [6] with an enhanced intelligent power.

The objective of the SemML project [15] is the creation of a stardardized markup language for semantic works. It seems that high expressive possibilities of SK-languages are to urge the authors of SemML project to update the goals of the project in the sense of considering some sublanguages of SK-languages and replacing some designations by the designations being more habitual for the programmers and Web designers.

6   Conclusions

The arguments stated above and numerous additional arguments set forth in the monograph [11] give serious grounds to conclude that the definition of the class of SK-languages can be interpreted as the first version of a broadly applicable and flexible Conceptual Metagrammar.

The theory of K-representations was developed as a tool for dealing with numerous questions of studying semantics of arbitrarily complex natural language texts: both sentences and discourses. Grasping the main ideas and methods of this theory requires considerably more time than it is necessary for starting to construct the formulas of the first-order logic. However, the efforts aimed at studying the foundations of the theory of K-representations would be highly rewarded. Independently on an application domain, a designer of a NL processing system will have a convenient tool for solving various problems.

References

1. Popescu, A.-M., Etzioni, O., Kautz, H.: Towards a Theory of Natural Language Interfaces to Databases. In: Proc. of the 8th Intern. Conf. on Intelligent User Interfaces, Miami, FL, pp. 149–157 (2003)
2. Kaufmann, E., Bernstein, A.: How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users? In: Aberer, K., et al. (eds.) ASWC/ISWC 2007. LNCS, vol. 4825, pp. 281–294. Springer, Heidelberg (2007)

3. Cimiano, P., Haase, P., Heizmann, J., Mantel, M.: ORAKEL: A Portable Natural Language Interface to Knowledge Bases. Technical Report, Institute AIFB, University of Karlsruhe, Germany (2007)
4. Frank, A., Krieger, H.-U., Xu, F., Uszkoreit, H., Crysmann, B., Jrg, B., Schaeffer, U.: Question Answering from Structured Knowledge Sources. J. of Applied Logic 5(1), 20–48 (2007)
5. Prince, V., Roche, M. (eds.): Information Retrieval in Biomedicine: Natural Language Processing for Knowledge Integration. IGI Global (2009)
6. Harrington, B., Clark, S.: ASKNet: Creating and Evaluating Large Scale Integrated Semantic Networks. In: Proceedings of the 2008 IEEE International Conference on Semantic Computing, pp. 166–173. IEEE Computer Society, Washington DC (2008)
7. Rindflesh, T.C., Kilicoglu, H., Fiszman, M., Roszemblat, G., Shin, D.: Semantic MEDLINE: An Advanced Information Management Application for Biomedicine. Information Services and Use, vol. 1, pp. 15–21. IOS Press (2011)
8. Wilks, Y., Brewster, C.: Natural Language Processing as a Foundation of the Semantic Web. Foundations and Trends in Web Science. Now Publ. Inc., Hanover (2006)
9. Fomichov, V.A.: The Formalization of Designing Natural Language Processing Systems. MAX Press, Moscow (2005) (in Russian)
10. Fomichov, V.A.: Theory of K-representations as a Source of an Advanced Language Platform for Semantic Web of a New Generation. In: Web Science Overlay J. On-line Proc. of the First Intern. Conference on Web Science, Athens, Greece, March 18-20 (2009), `http://journal.webscience.org/221/1/websci09_submission_128.pdf`
11. Fomichov, V.A.: Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms. Springer, Heidelberg (2010a)
12. Fomichov, V.A.: Theory of K-representations as a Comprehensive Formal Framework for Developing a Multilingual Semantic Web. Informatica. An International Journal of Computing and Informatics 34(3), 387–396 (2010b) (Slovenia)
13. Fomichov, V.A.: A Mathematical Model for Describing Structured Items of Conceptual Level. Informatica. An Intern. J. of Computing and Informatics 20(1), 5–32 (1996) (Slovenia)
14. Fomichov, V.A., Kirillov, A.V.: A Formal Model for Constructing Semantic Expansions of the Search Requests About the Achievements and Failures. In: Ramsay, A., Agre, G. (eds.) AIMSA 2012. LNCS, vol. 7557, pp. 296–304. Springer, Heidelberg (2012)
15. Harrington, B., Wojtinnik, P.-R.: Creating a Standardized Markup Language for Semantic Networks. In: Proceedings of the 2011 IEEE Fifth International Conference on Semantic Computing, pp. 279–282. IEEE Computer Society, Washington DC (2011)
16. Turnpenny, P.D., Ellard, S.: Emery's Elements of Medical Genetics, 12th edn. Elsevier Limited, Edinburgh (2005)

MOSAIC: A Cohesive Method for Orchestrating Discrete Analytics in a Distributed Model

Ransom Winder, Joseph Jubinski, John Prange, and Nathan Giles

MITRE Corporation, 300 Sentinel Drive, Annapolis Junction, MD 20701, USA
{rwinder,jjubinski,jprange,ngiles}@mitre.org

Abstract. Achieving an HLT analytic architecture that supports easy integration of new and legacy analytics is challenging given the independence of analytic development, the diversity of data modeling, and the need to avoid rework. Our solution is to separate input, artifacts, and results from execution by delineating different subcomponents including an inbound gateway, an executive, an analytic layer, an adapter layer, and a data bus. Using this design philosophy, MOSAIC is an architecture of replaceable subcomponents built to support workflows of loosely-coupled analytics bridged by a common data model.

Keywords: HLT, architecture, information extraction.

1   Introduction

There is a longstanding technical challenge in developing a capability to derive structured knowledge from the content of raw unstructured data using Human Language Technology (HLT) techniques as indicated by past attempts to achieve such a capability. Addressing this challenge is exacerbated by costs in time, resources, and effort. There is a wealth of narrowly focused, independently developed, and potentially evolving analytics that could contribute to the overall solution operating on large document corpora. One cannot mandate that independently developed analytics conform to a prescribed framework. Yet a system for workflows of such analytics is essential to make addressing higher-order problems manageable and repeatable.

A research environment is characterized by prototyping of analytics in workflows, debugging, interrogation of artifacts, repeatability, and parameter tuning. A system in this environment executing workflows of analytics faces several challenges. There must be support for legacy analytics so older efforts are not made instantly obsolete. Analytics must be seamlessly integrated with minimal rework. The system must be future proofed so that as technologies emerge for the system's different logically independent subcomponents new alternatives can be selected without a wholesale reengineering of the entire system. The system must be capable of processing and storing artifacts and knowledge generated from audio, image, and complex documents. This paper proposes an architecture, MOSAIC, to meet these challenges.

2   Background

The analytic integration problem presented in this work is not new. The DARPA TIPSTER Text Program was a 9-year multi-million dollar R&D effort to improve HLT for the handling of multilingual corpora for use within the intelligence process. Its first phase funded algorithms for Information Retrieval / Extraction, resulting in pervasive repeated functionality. The second phase sought to develop an architecture [1]. Despite a timetable of six months, years were required. Any success lay in making the architecture palatable for voluntary adoption [2].

From TIPSTER's Phase III emerged GATE (General Architecture for Text Engineering) [3]. GATE has evolved since 1995 and is widely used by European researchers. An important emphasis of GATE was a separation of data storage, execution, and visualization from the data structures and analytics. Integration in GATE is achieved by making use of standards of Java and XML to allow inter-analytic communication.

A third relevant HLT architecture is UIMA (Unstructured Information Management Architecture), a scalable integration platform for semantic analytics and search components. Developed by IBM, UIMA is a project at the Apache Software Foundation and has earned support in the HLT community. UIMA was meant to insulate analytic developers from the system concerns while allowing for tightly-coupled deployments to be delivered with the ease of service-oriented distributed deployments [4]. UIMA has a greater scalability than GATE when using UIMA AS to distribute analyses to operate in parallel as part of a single workflow.

A recent IBM success is Watson, the question answering system [5] capable of defeating human champions of Jeopardy, a striking feat of engineering. This was a multi-year intense research and development project undertaken by a core team of researchers that produced the DeepQA design architecture, where components that produce annotations or make assertions were implemented as UIMA annotators.

Recent work in developing an HLT architecture alternative [6] expressed the challenge of using existing HLT integration platforms in a research environment. In this project, Curator, the desire was to avoid a single HLT preprocessing framework as well as all-encompassing systems with steep learning curves such as is the case with GATE or UIMA. Rather, Curator was designed to directly support the use-case of diverse HLT analytics operating in concert. This articulates the scenario we raised.

3   Methods

3.1   Operating Restrictions and Architectural Subcomponents

MOSAIC grew from certain operating restrictions. Architectural subcomponents could not be costly in money or effort to use, the architecture should be flexible enough to overcome the obsolescence of any subcomponent, and it must be able to incorporate legacy analytics. The development of these analytics is beyond the control of users, and resiliency with respect to the "off the shelf" analytics is crucial. The architecture must tolerate independence between people with different roles with

respect to MOSAIC: developers, architects, and users. Developers are analytic crea-tors, architects are integrators into the architecture, and users make and execute workflows. MOSAIC must handle complex sequential and concurrent workflows with analytic modules as black boxes. Discrete analytics must not be tightly coupled to the workflow.

The *inbound gateway* handles the input stream of documents. Based on active workflow instances deployed by the executive, the inbound gateway will submit doc-uments to the data bus. Documents can be triaged here, such that only documents that match the active workflow instances' specifications move forward.

The *executive* for the system orchestrates all the user-specified behavior in the ex-ecution of a workflow. Workflows are capable of being deployed persistently and ad hoc. Workflow information is retained in the output, making results traceable and repeatable. Our constraints suggest a software framework targeted to analytic processing which handles crawling through documents, parsing the documents, and routing the documents to the appropriate analytics, treated as plug-ins to the system.

The *data bus* is responsible for collecting, managing, and indexing data in a distri-buted fashion, allowing search and retrieval of documents and artifacts. Artifacts can last the life of their workflow instance or be made persistent.

Analytics include any existing software packages used extensively for text to in-formation processing and smaller scale software crafted by subject matter experts. The system's flexibility means analytics not yet written can be added as plug-ins later.

Adapters are necessary for maintaining a common interchange format for the data to be passed between the analytics. A common interchange format can represent any-thing extracted or generated from the documents. This is a language that the adapters can interpret when translating data produced by one analytic for use by another.

Analytics typically produce raw formats of their own data objects. With an adapter layer, there is no expectation placed on analytic developers to write to the common interchange format or reengineer an existing analytic. Yet this does require that adap-ters be created for each raw format to convert output to the interchange format. For each particular input, another adapter must be written that will create data in this for-mat from data in the common interchange format. Analytics that share a raw model and format either input or output can use the same adapters.

Note, an analytic is concerned with the extraction or generation of artifacts from input, while the adapter is concerned with the conversion of one analytic format to another. Architects who are familiar with both the common model and individual raw analytic data models are the appropriate developers of adapters. This underscores the importance of having a separate adapter layer as this allows the analytics to be treated as buffered from the system integration and allows the common model to evolve without a direct impact on the analytics.

3.2   HLT Framework and Application

The case-study application involves workflows that operate on raw text, generating results mapped into a knowledge representation preserving annotation and prove-nance. Results are merged into a knowledge base generated from text source material.

Although content extraction analytics identify string sequences in text, it is desirable to generate a common knowledge representation from what is encoded in extent strings. Of greatest interest is representing entities and relationships. A triple store seems indicated. Because there is a need to apply annotation and provenance to more than just the participants of relationships—namely the relationships themselves—enhancing the triples into quads (named triples) is helpful. The TRIG syntax, a named graph format of the Notation 3 (N3) logic, was selected. This format consists of definitions of entity and relationship types and instances using four part statements that include a name, subject, predicate, and object.

The analytics are varied in this task of synthesizing knowledge from raw text. Some contain a full suite of capability, able to parse and tag the text, extract mentions of entities, co-reference them, and identify relationships between entities or events. Examples of full-stack analytics include BBN's Serif [7], LCC's Cicero [8], SRA's NetOwl [9], and Alias-i's LingPipe toolkit [10]. The first two are in use in the system.

There are many standalone analytics specialized to identify particular content, such as concepts, or information about the document itself or its author, such as demographic attributes. These analytics are often made independently as point solutions. MOSAIC allows them to be leveraged without insisting on a complete rewrite of their code, which is often infeasible and defeats the purpose of using the existing analytic.

There are analytics that rely on consuming the output of analytics in order to generate higher-order results. An example is METEOR [11], a system for capturing and reasoning over meeting, travel, and criminal events in raw text. METEOR takes as input the output of Serif, a properly processed document, and a library of lexicons.

This shows why a loosely-coupled architecture is ideal for a research environment. Treating METEOR as a discrete analytic allows for its continued development outside of the architecture without the burden of reworking it to be an integral part of a system, while providing it a context in which it can contribute to global result sets in a setting where its output is traceable and repeatable.

Options for MOSAIC subcomponents were assessed. For the executive, this survey included UIMA and GATE as well as maturing products (OpenPipeline), business workflow engines (BPEL), and scientific workflow engines (LONI or Ptolemy). Given the desire for loose integration and to avoid writing code to integrate, Ptolemy was chosen as it could treat the analytics and adapters as "black box" units.

For the data bus, the survey considered using a flat file system, open source content management systems, including Alfresco, and technologies for object data management, such as ObjectStore. The file system was judged to be too unstructured and unmanageable, and the object data management technology would impose too much rework on most analytics, which write file output. Therefore, Alfresco was chosen.

The development of adapters to migrate results into the common model is inescapable when aggregating results across analytics. HLT is a highly ambiguous domain, and content extraction models seldom have identical definitions, but with a rich common data model, the adaptation involves discovering alignment between results and the model, generating a mapping, and developing a parser for the formats.

4   Results

We examine system performance in a simple instance, using a workflow of three analytics: a custom-built text zoning analytic (Zoner), a representative content extraction analytic (Serif), and a custom-built analytic that merges and filters content (MF). Ten runs of a MOSAIC workflow on nine documents using these analytic elements were made, both with a MySQL database-backed store and a variant using F-Logic and OntoServer optimized by using caching and a buffered writer (Figure 1).

The mean overhead in orchestrating the analytics is negligible (50ms). The remaining overhead is in adaptation and the final write out of results. This is improved when switching to F-Logic. This switch incurs an additional cost to the MF analytic, but the overall mean execution time is reduced (30.1s for Fig. 1b vs. 61.8s for Fig. 1a). The standard deviation across the documents is lower for the F-Logic variant (0.4s vs. 5.2s for the version using MySQL). While adaptation and write out are not analytic work, they are necessary. If included in the overhead, it is 19.0% of overall execution. If orchestration alone is considered, overhead accounts for 0.2% of execution.
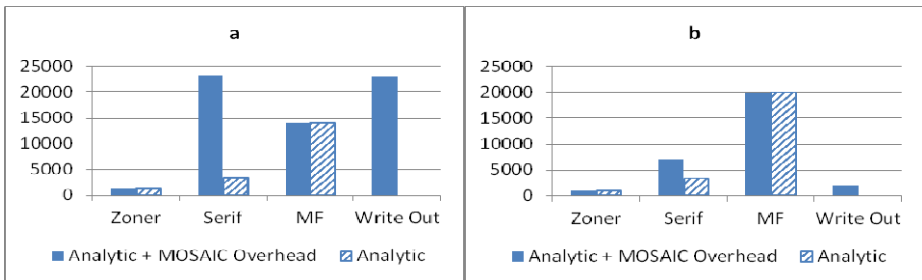

Fig. 1. Timing results (ms) for analytics and imposed overhead (orchestration, adaptation, and write out) using a MySQL-backed store (a) and a variant using F-Logic and OntoServer (b)

5   Discussion

MOSAIC was intended to address the needs of a research environment, but MOSAIC does not hinder the transition of workflow threads to production. It conformed to the requirements that the analytics be integrated seamlessly into a workflow that addresses larger-scope problems but without requiring integration-based rework on the analytics. The loosely-coupled nature of MOSAIC makes possible the rapid prototyping of the analytics in these workflows and permits substitution of subcomponents (i.e., executive, data bus) to allow for new technologies.

We have engineered an implementation of MOSAIC that embraces HLT analytics (text and speech) and supporting analytics of other domains (e.g. image processing, metadata analysis) across different workflows. At present, there are 16 HLT analytics and 5 supporting analytics in this implementation spanning 8 workflows geared toward solving larger problems within different genres of documents (textual, auditory,

image, and composites). The typical time to full integration for a new analytic that requires adapter development is improved over integration into our past efforts.

This adaptation is essential to the design of MOSAIC within the content extraction domain. Because the final results of the system are knowledge objects, these results need to have a cohesive representation despite the diversity of the models and formats of analytic output. Analytic developers cannot be expected to reengineer their analytics to fit our common representation, because it is often impossible to exert control over external analytic developers who did not model their analytics to the common representation and further it is not the role of the analytic developers to perform and maintain integration into a potentially evolving format. MOSAIC affords a division of labor such that it is the responsibility of MOSAIC integrators to perform this adaptation externally to the analytics. This does not remove the necessity for doing the adaptation work, but it does allow for the proper delineation of work roles such that this manner of integration is possible for analytics of disparate origins. There are domains of document processing (i.e., document decomposition, format and language conversion) which are not founded on producing knowledge results and have no requirements for adaptation, indicating MOSAIC could be used in these domains as is.

References

1. Altomari, P.J., Currier, P.A.: Focus of TIPSTER Phases I and II. In: Advances in Text Processing: TIPSTER Program Phase II, pp. 9–11. Morgan Kaufmann Publishers, Inc., San Francisco (April 1994-September 1996)
2. Grishman, R.: Building an Architecture: A CAWG Saga. In: Advances in Text Processing: TIPSTER Program Phase II, pp. 213–215. Morgan Kaufmann Publishers, Inc., San Francisco (1996)
3. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: An Architecture for Development of Robust HLT. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia, PA, pp. 168–175 (2002)
4. Ferrucci, D., Lally, A.: UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Natural Language Engineering 10(3-4), 327–348 (2004)
5. Ferrucci, D., et al.: Building Watson: an Overview of the DeepQA Project. AI Magazine 31(3), 59–79 (2010)
6. Clarke, J., Srikumar, V., Sammons, M., Roth, D.: An NLP Curator (or: How I Learned to Stop Worrying and Love NLP Pipelines). In: Proceedings of LREC 2012 (2012)
7. Boschee, E., Weischedel, R., Zamanian, A.: Automatic Information Extraction. In: Proceedings of the 2005 International Conference on Intelligence Analysis, pp. 2–4 (2005)
8. Surdeanu, M., Harabagiu, S.: Infrastructure for Open-domain Information Extraction. In: Proceedings of the Human Language Technology Conference, pp. 325–333 (2002)
9. SRA NetOwl,
   `http://www.sra.com/netowl/entity-extraction/features.php`
10. Alias-I, LingPipe 4.1.0, `http://alias-i.com/lingpipe`
11. Taylor, M., Carlson, L., Fontaine, S., Poisson, S.: Searching Semantic Resources for Complex Selectional Restrictions to Support Lexical Acquisition. In: Third International Conference on Advances in Semantic Processing, pp. 92–97 (2009)

Ranking Search Intents Underlying a Query

Yunqing Xia[1], Xiaoshi Zhong[1], Guoyu Tang[1], Junjun Wang[1], Qiang Zhou[1],
Thomas Fang Zheng[1], Qinan Hu[2], Sen Na[2], and Yaohai Huang[2]

[1] Tsinghua National Laboratory for Information Science and Technology, Department of
Computer Science and Technology, Tsinghua University, Beijing 100084, China
`{gytang,yqxia,xszhong,jjwang,zq-lxd,fzheng}@tsinghua.edu.cn`
[2] Canon Information Technology (Beijing) Co. Ltd., Beijing 100081, China
`{huqinan,nasen,huangyaohai}@canon-ib.com.cn`

Abstract. Observation on query log of search engine indicates that queries are usually ambiguous. Similar to document ranking, search intents should be ranked to facilitate information search. Previous work attempts to rank intents with merely relevance score. We argue that diversity is also important. In this work, unified models are proposed to rank intents underlying a query by combining relevance score and diversity degree, in which the latter is reflected by non-overlapping ratio of every intent and aggregated non-overlapping ratio of a set of intents. Three conclusions are drawn according to the experiment results. Firstly, diversity plays an important role in intent ranking. Secondly, URL is more effective than similarity in detecting unique subtopics. Thirdly, the aggregated non-overlapping ratio makes some contribution in similarity based intent ranking but little in URL based intent ranking.

Keywords: Intent ranking, relevance, diversity, non-overlapping ratio, aggregated non-overlapping ratio.

1 Introduction

Search engines receive billions of queries every day while more than 30 percent queries are ambiguous. The ambiguity can be classified into two types: (1) Meaning of the query cannot be determined. For example, in query "*bat*", one is difficult to know whether the query refers to a flying mammal or a tool for playing squash. (2) Facet of the query cannot be determined. For example, in query "*batman*", one cannot figure out which facet the user wants to know. Previous work ranks intents with relevance score and document similarity [1-2], which are insufficient. Observations disclose that intents usually overlap with each other. For example, *history of San Francisco* always mentions places and persons in the city. Search with "*San Francisco*" usually indicates three overlapping intents: *San Francisco history*, *San Francisco places* and *San Francisco people*. We argue the non-overlapping (i.e., unique) part amongst the intents plays a vital role in intent ranking.

In this work, unified models are proposed to rank intents underlying a query by combining relevance score and diversity degree. For the diversity degree, we propose the non-overlapping ratio to measure difference between intents. When calculating

cosine distance between two documents, we started from the term-based vector and further proposed the sense-based vector. Three conclusions are drawn from the experimental results. Firstly, diversity plays an important role in intent ranking. Secondly, URL is more effective than similarity in detecting unique subtopics. Thirdly, the aggregated non-overlapping ratio makes some contribution in similarity based intent ranking but little in URL based intent ranking.

The rest of this paper is organized as follows. In Section 2, we summarize related work. In Section 3, we report intent mining. We then present the non-overlapping ratio and intent ranking models in Section 4 and Section 5, respectively. Experiments and discussions are given in Section 6, and we conclude the paper in Section 7.

2   Related Work

Intent mining is a new research topic arising from NTCIR9 intent mining task [1]. To obtain the subtopic candidates, THUIR system uses Google, Bing, Baidu, Sogou, Youdao, Soso, Wikipedia and query log [2], which are proved helpful. Clustering on subtopic candidates is also used to find intents. For example, the Affinity Propagation algorithm is adopted in HITCSIR system to find intents [3]. In subtopic ranking, most NTCIR9 intent mining systems rely merely on relevance score [2-4]. Differently, we incorporated diversity into unified models for intent ranking.

Very recently, diversity has been explored by search engines to obtain diversified search results. For example, *uogTr* system applied the xQuAD framework for diversifying search results [5]. Some early diversification algorithms explore similarity functions to measure diversity [6]. Essential Pages algorithm was proposed to reduce information redundancy and returns Web pages that maximize coverage with respect to the input query [7]. This work is different as we propose unified intent ranking models considering both document relevance and intent overlap.

3   Discovering the Intents

In our intent mining system, intents are discovered from a set of subtopics, which are text strings reflecting certain aspects of the query.

3.1   Extracting Subtopics from Multiple Sources

For every query, we extract concepts (i.e., entries) within Wikipedia system using Wikipedia API. Subtopics are extracted based on concepts in four steps.

First, we extract subtopics from Wikipedia using disambiguation pages, redirect pages and table of content in content pages. Second, we extract more subtopics from user behavior data, e.g., query log, search engine recommendations, and search auto-completions. All the matched items are considered as subtopic candidates. Third, we induce subtopics from search results using Bayesian model [9] from the top 1000 search results. Finally, we assign the following rules to exclude the less likely subtopic candidates: (1) Candidates that are contained in the query are excluded; (2) Candidates that do not contain all concepts of the query are excluded.

3.2   Clustering Subtopics to Find Intents

We apply Affinity propagation (AP) clustering algorithm [10] to group the subtopic candidates. We revise the algorithm so that concepts are adopted.

We encounter a large proportion of named entities in the subtopic candidates. Consider two subtopic candidates: *furniture for small spaces New York*, *furniture for small spaces Los Angeles*. Obvious they refer to *furniture for small spaces* in two cities. For such cases, we adopt Freebase[1] to generalize subtopic candidates and associate named entities with the same ontology type.

4   Non-overlapping Ratio

Diversity is in fact reflected by non-overlapping (NOL) ratio, which is the ratio of non-overlapping parts over the overlapping parts within the intent. Consider an intent $I = \{t_1, t_2, ... t_N\}$, where t_i denotes a subtopic. Using subtopic t as a query, we obtain a set of search results $t \Rightarrow \{r_1, r_2, ..., r_M\}$ with a search engine, where r_j represents a search result. For Web search, we can further represent search result by the unique url string and document d: $r_j \equiv \{url_j, d_j\}$. Considering overlap, documents covered by an intent can be divided into unique part and common part.

We define *Non-Overlapping* (*NOL*) *ratio* of an intent as the ratio of unique part to the common part within the intent. Formally, given an intent I that covers a search result set $R = \{r_1, r_2, ..., r_\Phi\}$, we divide R into $R = R_{\text{uniq}} \cup R_{\text{comm}}$, where $R_{\text{uniq}} = \{r_{\text{uniq}}^1, r_{\text{uniq}}^2, ..., r_{\text{uniq}}^K\}$ and $R_{\text{comm}} = \{r_{\text{comm}}^1, r_{\text{comm}}^2, ..., r_{\text{comm}}^L\}$ represent the unique part and the remaining (common) part, respectively, and $K + L = \Phi$. NOL of intent I is calculated as follows.

$$ratio^{\text{NOL}} = \frac{\|R_{\text{uniq}}\| + \beta}{\|R_{\text{comm}}\| + \beta} \tag{1}$$

where β is set 1 to avoid the divided-by-zero error.

We designed two ways to count unique search results. In the first way, we simply compare the URL's of search results to determine uniqueness. In the second way, uniqueness is determined if it is not semantically similar to another search result. We adopt cosine distance in document similarity measuring based on vector space model.

5   Intent Ranking

We present two intent ranking models are designed based on NOL ratio.

5.1   Weighted NOL Ratio: Incorporating Relevance

We propose the weighted NOL (w-NOL) ratio that incorporates the relevance score in NOL ratio. Given relevance score w_{uniq}^k for search result r_{uniq}^k, and r_{comm}^l for w_{comm}^l. Eq.1 is revised as follows.

---

[1] Freebase: http://www.freebase.com/

$$ratio_w^{\mathrm{NOL}} = \frac{\sum_k w_{\mathrm{uniq}}^k + \beta}{\sum_l w_{\mathrm{comm}}^l + \beta} \qquad (2)$$

The relevance score is calculated with cosine distance. Finally, intents are ranked according to the weighted NOL (w-NOL) ratio.

5.2   NOL Ratio for a Set of Intents

Another solution views the intent ranking as an optimization problem that maximizes the aggregated NOL (ANOL) ratio of an intent set, which is the ratio of aggregated unique parts to the common parts within the intents. Formally, given an intent set Π that includes a set of intents $\Pi = \{I_1, I_2, ..., I_\Omega\}$ and a collection of search results $\widehat{R} = \{\widehat{r}_1, \widehat{r}_2, ..., \widehat{r}_\Sigma\}$ where search result set $\widehat{R}_v = \{\widehat{r}_{v,1}, \widehat{r}_{v,2}, ..., \widehat{r}_{v,\Phi_v}\}$ is covered by the intent I_v. By comparing the search results, we obtain $\widehat{R} = \widehat{R}_{\mathrm{uniq}} \cup \widehat{R}_{\mathrm{comm}}$, where $\widehat{R}_{\mathrm{uniq}} = \{\widehat{r}_{\mathrm{uniq}}^1, \widehat{r}_{\mathrm{uniq}}^2, ..., \widehat{r}_{\mathrm{uniq}}^Q\}$ represents the search results that are covered by only one intent, and $\widehat{R}_{\mathrm{comm}} = \{\widehat{r}_{\mathrm{comm}}^1, \widehat{r}_{\mathrm{comm}}^2, ..., \widehat{r}_{\mathrm{comm}}^O\}$ the rest search results, and $Q + O = \Sigma$. The ANOL ratio is calculated as follows.

$$ratio^{\mathrm{ANOL}} = \frac{\|\widehat{R}_{\mathrm{unique}}\| + \beta}{\|\widehat{R}_{\mathrm{common}}\| + \beta} \qquad (3)$$

Similar to w-NOL ratio, we obtain w-ANOL ratio by revising Eq.3 as follows.

$$ratio_w^{\mathrm{ANOL}} = \frac{\sum_e w_{\mathrm{unique}}^e + \beta}{\sum_g w_{\mathrm{common}}^g + \beta} \qquad (4)$$

Ranking intents with the w-ANOL ratio is an iterative process. It starts from the top ranked intent and ends with an intent list. Given n intents $\Pi^n = \{I_1, I_2, ..., I_n\}$ obtained in the n-th step, the $n+1$-th step seeks to find an intent I^* within the remaining intents that satisfies:

$$I^* = \mathrm{argmax}_{I \in \overline{\Pi^n}}\{ratio_w^{\mathrm{ANOL}}(\Pi^n + I)\} \qquad (5)$$

where $\overline{\Pi^n} = \Pi - \Pi^n$.

6   Evaluation

6.1   Experiment Setup

Dataset: NTCIR10 Intent-2 corpus (English) is used in the experimens[1].

Evaluation Metrics: We adopt two standard ranking metrics in the experiments:
-   *Normalized Discounted Cumulative Gain* on top N intents (nDCG@N) [1]. Performance score in the cutoff 10 results are evaluated.

- *Mean Average Precision* (MAP): MAP measures the mean of the average precision scores for each query.

Methods: The following intent ranking methods will be evaluated in our experiments.
- RIR: Intent ranking based merely on relevance.
- MIR: Intent ranking with MMR[4].
- UIR: Intent ranking according to w-NOL ratio based on URL.
- SIR: Intent ranking according to w-NOL ratio based on document similarity.
- UAIR: Intent ranking according to w-ANOL ratio based on URL.
- SAIR: Intent ranking according to w-ANOL ratio based on document similarity.

Experiment is conducted to justify contribution of the unified models. We set the similarity threshold value 0.8 in unique document determination.

6.2   Results and Discussions

Experimental results of the six methods are presented in Fig. 1. Three observations are made on the experimental results.
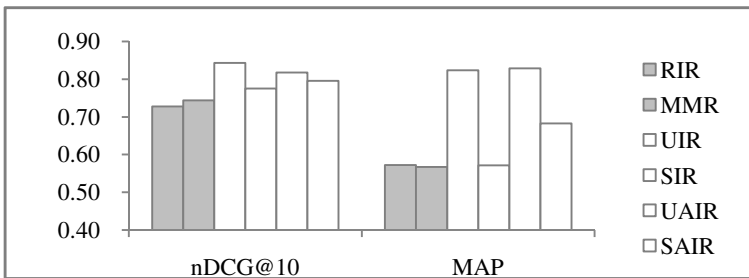

Fig. 1. Experimental results of the intent ranking methods

Firstly, we compare the NOL ratio based ranking methods (i.e., UIR, SIR, USIR and SAIR) against the traditional relevance based ranking methods (i.e., RIR and MIR). Seen from Fig.1, all the NOL ratio based ranking methods outperform the traditional methods significantly. It can be concluded that NOL ratio makes significant contribution to intent ranking. Secondly, we compare the four the NOL ratio based ranking methods (i.e., UIR, SIR, USIR and SAIR). Shown in Fig.1, the ANOL ratio based methods (i.e., USIR and SAIR) outperforms the NOL ratio based methods (i.e., UIR and SIR) on MAP. But on nDCG, there is no consistent outperformance. We conclude that the ANOL ratio tends to offer the accurate intents higher ranks while is not necessarily advantageous over NOL ratio in assigning the correct ranks. Thirdly, we compare the URL based intent ranking methods (i.e., UIR and USIR) and the similarity based methods (i.e., SIR and SAIR). Seen in Fig.1, the URL based methods outperform the similarity based methods consistently. We thus conclude that similarity do not contribute in detecting unique search results.

7   Conclusion

This paper seeks to prove that diversity is important in ranking intents underlying a query. Contributions of this work are summarized as follows. Firstly, diversity degree is incorporated in intent ranking. Secondly, non-overlapping ratio is proposed to calculate diversity degree of intent. Thirdly, intents are ranking with non-overlapping ratio in standalone manner and aggregating manner, respectively. Three conclusions are drawn according to the experimental results. First, diversity plays an important role in intent ranking. Second, URL is more effective than similarity in detecting unique subtopics. At last, the aggregated non-overlapping ratio makes some contribution in similarity based intent ranking but little in URL based intent ranking.

References

1. Song, R., Zhang, M., Sakai, T., Kato, M., Liu, Y., Sugimoto, M., Wang, Q., Orii, N.: Overview of the NTCIR-9 INTENT Task. In: Proc. of NTCIR-9 Workshop Meeting, Tokyo, Japan, December 6-9, pp. 82–104 (2011)
2. Xue, Y., Chen, F., Zhu, T., Wang, C., Li, Z., Liu, Y., Zhang, M., Jin, Y., Ma, S.: THUIR at NTCIR-9 INTENT Task. In: Proc. of NTCIR-9, Tokyo, Japan, December 6-9 (2011)
3. Song, W., Zhang, Y., Gao, H., Liu, T., Li, S.: HITSCIR System in NTCIR-9 Subtopic Mining Task. In: Proc. of NTCIR-9, Tokyo, Japan, December 6-9 (2011)
4. Han, J., Wang, Q., Orii, N., Dou, Z., Sakai, T., Song, R.: Microsoft Research Asia at the NTCIR-9 Intent Task. In: Proc. of NTCIR-9, Tokyo, Japan, December 6-9 (2011)
5. Santos, R.L.T., Macdonald, C., Ounis, I.: University of Glasgow at the NTCIR-9 Intent task: Experiments with Terrier on subtopic mining and document ranking. In: Proc. of NTCIR-9, Tokyo, Japan, December 6-9 (2011)
6. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proc. of SIGIR 1998, Melbourne, Australia, pp. 335–336 (1998)
7. Swaminathan, A., Mathew, C.V., Kirovski, D.: Essential Pages. In: Proc. of WI 2009, Milan, Italy, pp. 173–182 (2009)
8. Santamaría, C., Gonzalo, J., Artiles, J.: Wikipedia as sense inventory to improve diversity in web search results. In: Proc. of ACL 2010, Uppsala, Sweden, pp. 1357–1366 (2010)
9. Brody, S., Lapata, M.: Bayesian word sense induction. In: Proc. of EACL 2009, pp. 103–111 (2009)
10. Dueck, D.: Affinity Propagation: Clustering Data by Passing Messages. University of Toronto Ph.D. thesis (June 2009)

Linguistic Sentiment Features for Newspaper Opinion Mining

Thomas Scholz and Stefan Conrad

Heinrich-Heine-University, Institute of Computer Science, Düsseldorf, Germany
{scholz,conrad}@cs.uni-duesseldorf.de

Abstract. The sentiment in news articles is not created only through single words, also linguistic factors, which are invoked by different contexts, influence the opinion-bearing words. In this paper, we apply various commonly used approaches for sentiment analysis and expand research by analysing semantic features and their influence to the sentiment. We use a machine learning approach to learn from these features/influences and to classify the resulting sentiment. The evaluation is performed on two datasets containing over 4,000 German news articles and illustrates that this technique can increase the performance.

Keywords: Opinion Mining, Sentiment Analysis, Media Response Analysis.

1   Introduction

Every day, many news texts are published and distributed over the internet (uploaded newspaper articles, news from online portals). They contain potentially valuable opinions. Many organisations analyse the polarity of sentiment in news items which talk about them. How is the media image about company XY? Is the sentiment changing after the last advertising campaign? For instance, a Media Response Analysis (MRA) answers these questions [12]. In a MRA, several media analysts have to read the collected news, select relevant statements from the articles and assign a sentiment for each statement. This means in effect, a MRA needs a big human effort. At the same time, the internet contains more and more potentially relevant articles. As a consequence, media monitoring services require more machine-aided methods. Opinions are not stated so clearly in newspaper articles [1]. In the news, some special features are important for the sentiment, so that an only-word-based method cannot solve this problem.

Formal Task Definition: *Given a statement s which consists of the words w_i with $i \in \{1, ..., s_n\}$. The task is to find the polarity of sentiment y for the statement s:*

$$f : s = (w_1, ..., w_{s_n}) \mapsto y \in \{pos, neg\} \tag{1}$$

2   Related Work

Research in Opinion Mining is far-reaching [7], however the most techniques tackle this problem in the domain of customer reviews [7]. Many approaches for Opinion Mining in reviews collect sentiment-bearing words [6]. There are methods [4] which try to handle linguistic or contextual sentiment such as negations. The negation as the maybe most important linguistic factor is often treated by heuristic rules [4], which reverse the polarity of sentiment words. Interesting techniques for the effects of negations have been introduced by Jia et al. [5]. Here, the scope of negations are derived from different rules. In addition, we are interested in a linguistic and grammatical context as in Zhou et al. [13]. They show that conjunctions can be used to avoid ambiguities within sentences. In the news domain, many approaches on this topic only work with reported speech objects [1]. News articles are less subjective [1], but quotations in newspaper articles are often the place where more subjective text and opinions can be found [1]. However, only opinions, which are part of a reported speech object, can be analysed by this method. An analysis [9] shows that in a MRA less than 22% of the opinion-bearing text contain quoted text and only in less than 5% the area of quoted text is larger than 50% of the whole relevant opinion.

3   Determination of Sentiment Polarity

Our approach calculates four basic sentiment features (**Basic Sentiment Features** α) first. These features are based on the four word categories adverbs, adjectives, nouns, and verbs, which are the most important word classes for the polarity of sentiment [8]. We use existing methods such as chi-square [6], the PMI-method [3,6], the entropy-based method [11], the method of information gain [11], and the German sentiment lexicon SentiWS [8] for the weighting of the polarity (our sentiment score σ). We compute four sentiment features for one statement (**Basic Sentiment Features** α). Every feature is the average of the sentiment scores in one category: The first feature is the average of the scores of all the statement's adjectives ($f_{\alpha_1}(s) = \sigma_{Adj}(s)$), the second of all nouns ($f_{\alpha_2}(s) = \sigma_{No}(s)$), and so on.

$$\sigma_{cat}(s) = \frac{1}{|s_{cat}|} \sum_{w \in s_{cat}} \sigma_{method}(w) \qquad (2)$$

Here, s_{cat} are only the words in statement s which belong to one of the four important categories (adjectives, nouns, verbs, and adverbs) and σ_{method} is one of the five word based methods.

4   Linguistic and Contextual Features

4.1   Two Techniques for the Effect Measurement

The first technique only measures, whether or not the linguistic effects are present in a given statement and stores it as one feature for every aspect

(**Linguistic Effect Features** β). The second technique tries to capture an area of this effect and it takes the sentiment of the area as the feature value of this aspect (resulting in **Linguistically Influenced Sentiment Features** γ). The feature value is the sum of the sentiment of the influenced words. We implement techniques from Jia et al. [5], who are trying to capture different effect areas for negations. We adapt their *candidate scope* [5] and *delimiter rules* [5] using static and dynamic delimiters for the German language and expand them also for our non negation features: The static *delimiters* [5] remove themselves and all words after them from the scope. Static *delimiters* are words such as "because", "when" or "hence" [5]. A *conditional delimiter* [5] becomes a delimiter if it has the correct POS-tag, is inside a negation scope, and leads to opinion-bearing words. Examples are words such as "who", "where" or "like". In addition, we have designed a second method which creates a scope around an effect word. All words in the scope have a smaller distance to all other effect words (in number of words between them).

4.2   Calculation of the Features

The sentiment of words can change depending on whether the statements concern persons or organisations. So, the first two features represesent the proportion of persons and organisations: In equation 3 for the first two β features, $p(s)$ and $o(s)$ are the number of persons and organisations, respectively, in the statement s. For the two type γ features, P_w and O_w are the set of words which belongs to persons' and organisations' scope (second method, cf. section 4.1), respectively.

$$f_{\beta_1}(s) = \frac{p(s)}{p(s) + o(s)} \qquad f_{\beta_2}(s) = \frac{o(s)}{p(s) + o(s)} \tag{3}$$

$$f_{\gamma_1}(s) = \sum_{w \in P_w} \sigma(w) \qquad f_{\gamma_2}(s) = \sum_{w \in O_w} \sigma(w) \tag{4}$$

The negation feature shows, whenever a negation is present in statement s. N_w are the affected words. At this point, the area of affected words is determined by the *candidate scope* [5] and *delimiter rules* [5].

$$f_{\beta_3}(s) = \begin{cases} 1.0 \text{ if } \exists w \in s : w \text{ is a negation} \\ 0.0 \text{ otherwise} \end{cases} \qquad f_{\gamma_3}(s) = \sum_{w \in N_w} \sigma(w) \tag{5}$$

The use of conjunctions can also indicate a polarity. We create a test data of 1,600 statements, collect the conjunctions and associate them with a sentiment value ν_c by their appearance in positive and negative statements. Table 1 (left) shows the different conjunctions and their value to influence the sentiment. The type β feature for conjunctions is the sum of all sentiment values ν_c of all conjunctions C_s of the statement s. The conjunction influenced words are C_w. The scope is

Table 1. Left: Conjunctions and sentiment value. Right: Hedging auxiliary verbs.

word	ν_c	word	ν_c	word	ν_c	word	ν_c
whereas	-0.5	as well	1.0	but	-1.0	or	0.5
however	-0.5	though	-1.0	and	1.0	by	1.0

can	may	could	might
would	shall	should	ought to
will	must		

determined by the *candidate scope* [5] and *delimiter rules* [5], but only words after the conjunction are concerned because the conjunction itself is a delimiter. The multiplication with ν_c indicates which type of conjunction influences the affected words. If the conjunction expresses a contrast (e.g. "but" with $\nu_c = -1.0$), the sentiment of the words will be inverted.

$$f_{\beta_4}(s) = \frac{\sum_{c \in C_s} \nu_c}{|C_s|} \qquad f_{\gamma_4}(s) = \sum_{w \in C_w} \nu_c * \sigma(w) \qquad (6)$$

A short part of quoted text can be a hint for irony in written texts [2] and a long part can stand for a reported speech object. As a result, a machine learning approach can better differentiate between irony and reported statements, if the length and the affected words of quoted text are measured. $q(s)$ is the part of a statement s, which appears in quotation marks. $l(x)$ is the length (in characters) of a text x. Q_w are the words inside a quotation.

$$f_{\beta_5}(s) = \frac{l(q(s))}{l(s)} \qquad f_{\gamma_5}(s) = \sum_{w \in Q_w} \sigma(w) \qquad (7)$$

Modal verbs like "can" or "would" can weaken the strength of the polarity. The full list of auxiliary verbs for hedging expressions is shown in table 1 (right). The method counts how often full verbs are influenced by hedging expressions $h(s)$ in comparison to all full verbs $v(s)$. H_w is the set of words affected by hedging. Here again, the *candidate scope* [5] and *delimiter rules* [5] are used.

$$f_{\beta_6}(s) = \frac{h(s)}{v(s)} \qquad f_{\gamma_6}(s) = \sum_{w \in H_w} \sigma(w) \qquad (8)$$

4.3 Machine Learning Technique for Sentiment Classification

For the classification, we use a SVM (Rapidminer[1] standard implementation). The SVM receives the feature sets β and γ as input values for learning, as well as it obtains the **Basic Sentiment Features** α. In this way, our machine learning approach is able to learn from the sentiment features and the linguistic features.

---

[1] Rapid-I: `http://rapid-i.com/`

5   Evaluation

We evaluate our approach on two different datasets: The first corpus, called
Finance, represents a real MRA about a financial service provider. It contains
5,500 statements (2,750 are positive, 2,750 are negative) from 3,452 different
news articles. The second dataset is the **pressrelations** dataset [10]. We use
approx. 30% of the dataset to construct a sentiment dictionary. This means that
1,600 statements (800 are positive, 800 are negative) are used for Finance and
308 statements for the pressrelations dataset. The sentiment dictionaries contain
words which are weighted by the methods explained in section 3. We use 20% of
the remaining set to train a classification model. The results are depicted in table
2 and show that the features β and γ improve sentiment allocation. The features
increased performance of all methods, except the information gain method on
pressrelations. However, in all other cases, the methods achieved the best results
by using all features. SentiWS, as the dictionary based approach, got the highest
improvement (over 7% on finance and over 14% on pressrelations). The entropy-
based method with all features got the highest accuracy with 75.28% on Finance,
which is an improvement of over 5% to the baseline.

Table 2. Results of the linguistic features

Method	Finance dataset				pressrelations dataset			
	α	$\alpha+\beta$	$\alpha+\gamma$	all	α	$\alpha+\beta$	$\alpha+\gamma$	all
SentiWS	0.6036	0.6590	0.6311	**0.6792**	0.5526	0.5604	0.615	**0.6943**
PMI	0.6174	0.6586	0.6317	**0.6881**	0.6245	0.6057	0.634	**0.6887**
χ^2	0.6872	0.7071	0.6981	**0.7234**	0.6453	0.6453	0.6717	**0.6868**
Entropy	0.7006	0.7221	0.7428	**0.7528**	0.6642	0.6604	0.6774	**0.6943**
Information Gain	0.6955	0.7186	0.7243	**0.7349**	**0.6912**	0.6761	0.6811	0.6828

By comparing all results, the influence of feature set β seems to be bigger than
the influence of feature set γ on Finance, while it is the other way around on the
pressrelations dataset. The reason for this is the nature of the two domains. The
political texts are more complicated so that a deeper analysis, which exploits
values of the influenced sentiment-bearing words, provides more benefit. Never-
theless, except the for information gain method, the combination of all linguistic
features achieved an increase to the baselines of at least over 3%.

6   Conclusion

In conclusion, linguistic features are very useful for Opinion Mining in newspa-
per articles. The evaluation shows that the linguistic features can be integrated
into existing solutions and thereby improve the computation of sentiment. The
improvement is especially large and therefore interesting for dictionary based
approaches. Moreover, this approach achieved high accuracies of over 70% and
in one case an accuracy of over 75%.

References

1. Balahur, A., Steinberger, R., Kabadjov, M., Zavarella, V., van der Goot, E., Halkia, M., Pouliquen, B., Belyaeva, J.: Sentiment analysis in the news. In: Proc. of the 7th Intl. Conf. on Language Resources and Evaluation, LREC 2010 (2010)
2. Carvalho, P., Sarmento, L., Silva, M.J., de Oliveira, E.: Clues for detecting irony in user-generated contents: oh..!! it's "so easy";-). In: Proc. of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion, TSA 2009, pp. 53–56 (2009)
3. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. In: Proc. of the 27th Annual Meeting on Association for Computational Linguistics, ACL 1989, pp. 76–83 (1989)
4. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Proc. of the Intl. Conf. on Web Search and Web Data Mining, WSDM 2008, pp. 231–240 (2008)
5. Jia, L., Yu, C., Meng, W.: The effect of negation on sentiment analysis and retrieval effectiveness. In: Proc. of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, pp. 1827–1830 (2009)
6. Kaji, N., Kitsuregawa, M.: Building lexicon for sentiment analysis from massive collection of html documents. In: Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL (2007)
7. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
8. Remus, R., Quasthoff, U., Heyer, G.: SentiWS – a publicly available german-language resource for sentiment analysis. In: Proc. of the 7th Intl. Conf. on Language Resources and Evaluation, LREC 2010 (2010)
9. Scholz, T., Conrad, S.: Integrating viewpoints into newspaper opinion mining for a media response analysis. In: Proc. of the 11th Conf. on Natural Language Processing, KONVENS 2012 (2012)
10. Scholz, T., Conrad, S., Hillekamps, L.: Opinion mining on a german corpus of a media response analysis. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2012. LNCS, vol. 7499, pp. 39–46. Springer, Heidelberg (2012)
11. Scholz, T., Conrad, S., Wolters, I.: Comparing different methods for opinion mining in newspaper articles. In: Bouma, G., Ittoo, A., Métais, E., Wortmann, H. (eds.) NLDB 2012. LNCS, vol. 7337, pp. 259–264. Springer, Heidelberg (2012)
12. Watson, T., Noble, P.: Evaluating public relations: a best practice guide to public relations planning, research & evaluation. PR in practice series, ch. 6, pp. 107–138. Kogan Page (2007)
13. Zhou, L., Li, B., Gao, W., Wei, Z., Wong, K.-F.: Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In: Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 162–171 (2011)

Text Classification of Technical Papers Based on Text Segmentation

Thien Hai Nguyen and Kiyoaki Shirai

Japan Advanced Institute of Science and Technology
{nhthien,kshirai}@jaist.ac.jp

Abstract. The goal of this research is to design a multi-label classification model which determines the research topics of a given technical paper. Based on the idea that papers are well organized and some parts of papers are more important than others for text classification, segments such as title, abstract, introduction and conclusion are intensively used in text representation. In addition, new features called Title Bi-Gram and Title SigNoun are used to improve the performance. The results of the experiments indicate that feature selection based on text segmentation and these two features are effective. Furthermore, we proposed a new model for text classification based on the structure of papers, called Back-off model, which achieves 60.45% Exact Match Ratio and 68.75% F-measure. It was also shown that Back-off model outperformed two existing methods, ML-kNN and Binary Approach.

Keywords: Text Classification, Multi-label Classification, Text Segmentation, Supervised Learning.

1 Introduction

In many research fields, a lot of papers are published every year. When researchers look for technical papers by a search engine, only papers including user's keywords are retrieved, and some of them might be irrelevant to the research topics that users want to know. Therefore, a survey of past researches is hard and difficult. Automatic identification of the research topics of the technical papers would be helpful for the survey. It is a kind of text classification problem.

Our goal is to design an effective model which determines the categories of a given technical paper about natural language processing. In our approach, the model will consider the text segments in the paper. Several models with different feature sets from different segments are trained and combined. Furthermore, new features associated with the title of the paper are introduced.

2 Background

Text classification has a long history. Many techniques have been studied to improve the performance. The commonly used text representation is bag-of-words [1]. Not words but phrases, word sequences or N-grams [2] are sometimes

used. Most of them focused on words or N-grams extracted from the whole document with feature selection or feature weighting scheme. Some of the previous work aimed at the integration of document contents and citation structure [3] [4].

Nomoto supposes the structure of the document as follows: the nucleus appears at the beginning of the text, followed by any number of supplementary adjuncts [5]. Then keywords for text classification are extracted only from the nucleus. Identification of nucleus and adjuncts is as a kind of text segmentation, but our text segmentation is fit for technical papers.

Larkey proposed a method to extract words only from the title, abstract, the first twenty lines of summary and the section containing the claims of novelty for a patent categorization application [6]. His method is similar to our research, but he classifies the patent documents, not technical papers. Furthermore, we proposed a novel method called back-off model as described in Subsection 4.4.

There are many approaches for multi-label classification. However, they can be categorized into two groups: problem transformation and algorithm adaptation [7]. The former group is based on any algorithms for single-label classification. They transform the multi-label classification task into one or more single-label classification. On the other hand, the latter group extends traditional learning algorithms to deal with multi-label data directly.

3   Dataset

We collect technical papers in proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) from 2000 to 2011. To determine the categories (research topics) of the papers, we first refer the category list used for paper submission to the Language Resources and Evaluation Conference (LREC). Categories are coarse grained research topics such as syntactic parsing, semantic analysis, machine translation and so on. Categories for each paper in the collection are annotated by authors. The total number of papers in the collection is 1,972, while the total number of categories is 38. The average number of the categories per a paper is 1.144. Our dataset is available on the git repository [1].

4   Multi-label Classification of Technical Papers

4.1   Text Segmentation

As the preprocessing of text classification, the following segments in the paper are automatically identified: title, author information (authors' names, affiliations, e-mail addresses etc.), abstract, introduction, conclusion and reference. Title is gotten from the database of papers shown in Section 3. A segment from the beginning of the paper to abstract is supposed to be an author information section. Abstract, introduction, conclusion and reference sections are identified by keywords in the papers.

---

[1] `https://github.com/nhthien/CorpusACL`

4.2   Title Feature

In addition to the ordinary bag-of-word features, we propose new types of feature derived from the title of the paper. Words in the title seem the most important for paper classification. However, not all words in the title may be effective features. In this paper, 'Title Bi-Gram' and 'Title SigNoun' are proposed to overcome this problem. 'Title Bi-Gram' is defined as bi-gram in noun phrases in the title. The motivation of 'Title Bi-Gram' feature is that the noun phrases in the title represent research topic clearly. Another title feature is 'Title SigNoun', which is defined as a noun in a head NP and a noun in a prepositional phrase (PP). This feature is represented in the form of '$p+n$', where n and p is a noun in PP and a head preposition of PP, respectively. The motivation of 'Title SigNoun' feature is that not only the nouns in the head NP but also in some cases the words in the prepositional phrase describe topics of papers. For example, a prepositional phrase "for information retrieval" strongly indicates that the paper tends to belong to "Information Retrieval" category, while "with bilingual lexicon" might not be helpful in identifying topics of papers. The feature represented as the combination of the noun with the preposition, such as 'for+retrieval' or 'with+lexicon', enables us to distinguish effective and ineffective prepositional phrases. For example, from the title "Annotating and Recognising Named Entities in Clinical Notes", 'Named Entities' and 'Clinical Notes' are extracted as Title Bi-Gram, while 'Named', 'Entities' and 'in+Notes' are extracted as Title SigNoun feature.

4.3   Feature Selection

We propose a method of feature selection based on the segments of the paper. Only words in useful segments such as title, abstract, introduction and conclusion are selected as features. We consider the five feature sets as follows:

1. The whole content of paper: all of the words will be selected as features.
2. Words in title, abstract, introduction and conclusion (TAIC).
3. Words in TAIC and Title Bi-Gram.
4. Words in TAIC and Title SigNoun.
5. Words in TAIC, Title Bi-Gram and Title SigNoun.

4.4   Classification Models

As discussed in Section 2, there are two approaches for multi-label classification: algorithm adaptation and problem transformation. We choose ML-kNN as the former and binary approach as the latter. ML-kNN [8] is a multi-label lazy learning approach. We used MULAN [9] as ML-kNN implementation in our experiments. Binary Approach [7] is a model that determines categories from results of $|C|$ binary classifiers for each different label, where C is a label set. We used LibSVM [10] with linear kernel to train each binary classifier.

Based on the structure of papers, we propose a new model 'back-off model' derived from the binary approach. To improve the precision, only categories

with high posterior probability from different perspectives are selected. Here the perspectives are binary approach methods with different feature sets. Figure 1 shows an architecture of back-off model. At first, a model with a basic feature set judges categories for the paper. The basic feature set is a set of words in the title with Title Bi-Gram and/or Title SigNoun feature [2]. The results of model 1 are a list of categories with their posterior probabilities $\{(C_i, P_{i1})\}$. The system outputs categories C_i where P_{i1} are greater than a threshold T_1. When no class is chosen, model 2 using words in the abstract as well as basic features is applied. Similarly, model 3 (using words in introduction as well) and model 4 (using words in conclusion as well) are applied in turn. When no class is chosen by model 4, all categories whose probabilities P_{ik} are greater than 0.5 are chosen. If no P_{ik} is greater than 0.5, the system chooses one class with the highest probability. The threshold T_k for the model k is set smaller than that of the previous step. We investigate several sets of thresholds in the experiments in Section 5.

Fig. 1. Architecture of Back-off Model

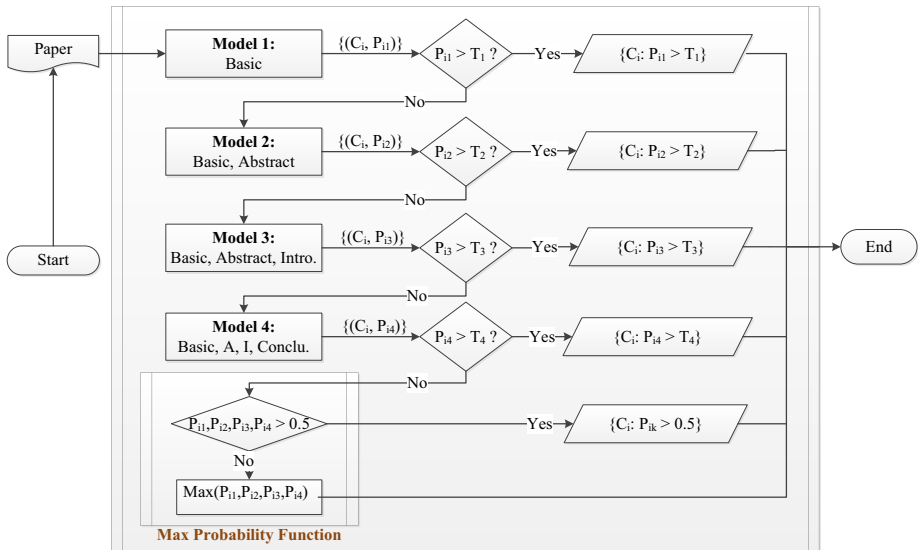## 5   Evaluation

The proposed methods are evaluated by 10-fold cross validation on the collection of the papers described in Section 3. We used exact match ratio (EMR), accuracy,

---

[2] Three basic feature sets were investigated: Title + Title Bi-Gram (BF_1), Title + Title SigNoun (BF_2) and Title + Title Bi-gram + Title SigNoun (BF_3). In our experiments, BF_1 achieved the best.

precision and recall as the instance-based metrics [3], and micro-Precision, micro-Recall, micro-F, macro-Precision, macro-Recall, and macro-F as the category-based metrics [4]. Although we have evaluated various feature sets and parameters for ML-kNN, binary approach and back-off model, only some of the results will be shown in this paper due to the lack of space.

Table 1 reveals results of binary approach. It shows that using feature selection by text segmentation gives better results than using all content of the paper[5]. In addition, combining Title Bi-Gram and Title SigNoun improves the performance[6]. Table 2 shows the results of back-off model with some combinations of thresholds $T_1 \sim T_4$. We found that the performance of back-off model did not highly depend on the thresholds.

Table 1. Results of Binary Approach

Feature Set	Instance-based Metrics				Category-based Metrics					
	EMR	A	P	R	Mi-P	Mi-R	Mi-F	Ma-P	Ma-R	Ma-F
All	46.55	59.00	62.51	68.89	57.57	67.44	62.10	49.22	58.83	53.55
TAIC	51.72	61.80	65.10	68.93	62.15	67.26	64.58	55.59	59.35	56.74
TAIC + Title SigNoun	52.84	62.95	66.27	69.98	63.40	68.41	65.79	56.52	59.62	57.79
TAIC + Title Bi-Gram	52.94	63.37	66.90	70.57	63.91	68.89	66.29	57.59	61.27	58.59
TAIC + Title Bi-Gram + Title SigNoun	**53.80**	**64.05**	**67.38**	**71.20**	**64.57**	**69.65**	**66.99**	**58.17**	**61.36**	**59.72**

Table 2. Best Results of Back-off Model

Thresholds T_1-T_2-T_3-T_4	Instance-based Metrics				Category-based Metrics					
	EMR	A	P	R	Mi-P	Mi-R	Mi-F	Ma-P	Ma-R	Ma-F
80-80-50-50	60.04	67.21	72.01	**69.75**	70.20	67.39	**68.76**	65.66	**59.90**	61.97
80-80-70-50	60.14	67.09	71.97	69.32	70.43	66.91	68.61	65.80	59.43	61.73
80-80-80-50	**60.45**	**67.25**	**72.07**	69.44	**70.58**	67.04	68.75	**66.33**	59.85	**62.16**

To compare the performance of ML-kNN, binary approach and back-off model, the highest values among various feature sets and parameters for three models are shown in Figure 2. It indicates that ML-kNN performs much worse than binary approach and back-off model on all metrics. Binary approach method outperformed back-off model on recall, micro-Recall and macro-Recall metrics. In contrast, back-off model tends to achieve better results on EMR, accuracy, precision, micro-Precision, macro-Precision, micro-F and macro-F. Therefore, back-off model is the best among three approaches.

---

[3] EMR is a proportion of instances (papers) where the gold and predicted set of categories are exactly same. While others evaluate the predicted categories for individual instances.

[4] They are averages of prediction of individual categories.

[5] Differences between All and TAIC are verified by a statistical test called randomization test of paired sample [11]. They are statistically significant.

[6] Differences between models with and without Title Bi-Gram/SigNoun were statistically significant.
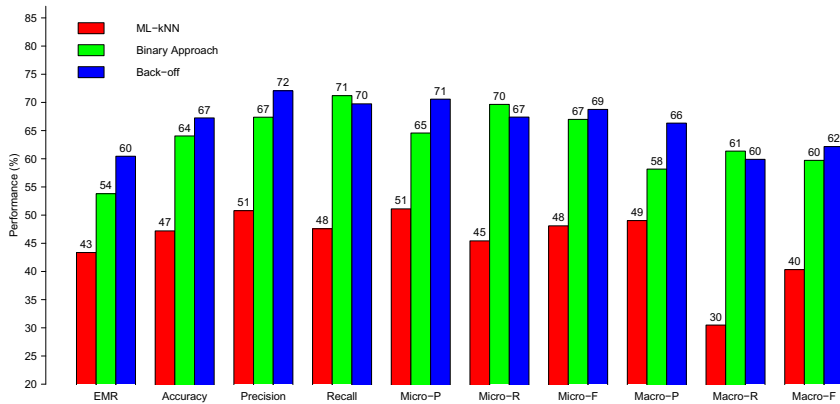
Fig. 2. Best Performance of Three Models

6   Conclusion

To identify research topics of papers, we proposed a feature selection method based on the structure of the paper and new features derived from the title. We also proposed back-off model, which combines classifiers with different feature sets from different segments of the papers. Experimental results indicate that our methods are effective for text categorization of technical papers. In the future, we will explore more effective methods of feature selection and feature weighting to improve the accuracy of text classification.

References

1. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34(1), 1–47 (2002)
2. Rahmoun, A., Elberrichi, Z.: Experimenting n-grams in text categorization. Int. Arab J. Inf. Technol., 377–385 (2007)
3. Cao, M.D., Gao, X.: Combining contents and citations for scientific document classification. In: Australian Conference on Artificial Intelligence, pp. 143–152 (2005)
4. Zhang, M., Gao, X., Cao, M.D., Ma, Y.: Modelling citation networks for improving scientific paper classification performance. In: Yang, Q., Webb, G. (eds.) PRICAI 2006. LNCS (LNAI), vol. 4099, pp. 413–422. Springer, Heidelberg (2006)
5. Nomoto, T., Matsumoto, Y.: Exploiting text structure for topic identification. In: Proceedings of the 4th Workshop on Very Large Corpora, pp. 101–112 (1996)
6. Larkey, L.S.: A patent search and classification system. In: Proceedings of the Fourth ACM Conference on Digital Libraries, DL 1999, pp. 179–187. ACM, New York (1999)
7. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer US (2010)

8. Zhang, M.L., Zhou, Z.H.: Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognition 40(7), 2038–2048 (2007)

9. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: A java library for multi-label learning. Journal of Machine Learning Research 12, 2411–2414 (2011)

10. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

11. Morgan, W.: Statistical hypothesis tests for NLP, http://cs.stanford.edu/people/wmorgan/sigtest.pdf

Product Features Categorization Using Constrained Spectral Clustering

Sheng Huang, Zhendong Niu, and Yulong Shi

School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
huangsheng2009@gmail.com,{zniu,sylbit}@bit.edu.cn

Abstract. Opinion mining has increasingly become a valuable practice to grasp public opinions towards various products and related features. However, for the same feature, people may express it using different but related words and phrases. It is helpful to categorize these words and phrases, which are domain synonyms, under the same feature group to produce an effective opinion summary. In this paper, we propose a novel semi-supervised product features categorization strategy using constrained spectral clustering. Different from existing methods that cluster product features using lexical and distributional similarities, we exploit the morphological and contextual characteristics between product features as prior constraints knowledge to enhance the categorizing process. Experimental evaluation on real-life dataset demonstrates that our proposed method achieves better results compared with the baselines.

Keywords: Product Features Categorization, Constrained Spectral Clustering, Constraint Propagation, Opinion Mining.

1 Introduction

With the exponential growth of online reviews, it becomes practical to automatically analyze large-scale customer opinions towards various products and related features. Fine-grained opinion mining has been emerging as a valuable research practice in the fields of natural language processing, machine learning and text mining. However, reviewers may use different words and phrases to express the same product feature, e.g. "image", "photo" and "picture" are feature expressions referring to the "photo" feature in camera domain, it becomes critical to categorize these feature expressions in feature-oriented opinion mining and summarization.

In this paper, we propose a novel semi-supervised product features categorization strategy using constrained spectral clustering. We define and extract the morphological and contextual constraints between feature expressions as prior knowledge. The constraint propagation is employed to spread the local pair-wise constraints throughout all feature expressions. The propagated constraints are incorporated into the spectral clustering to categorize the feature expressions into meaningful features. Our strategy consists of three components: constraints definition and extraction, constraint propagation and constrained spectral clustering.

2   Related Work

Existing product features categorization studies are mainly focused on clustering product features based on their lexical and distributional similarities [1-5]. [1] employed WordNet to check if any synonym set exists among the features. [2] proposed an unsupervised categorization method that employed two latent semantic association models to group context words into concepts and categorize product features respectively. [3] proposed to cluster product features and opinion words simultaneously and iteratively by fusing both content and sentiment link information. [4] presented a method to map discovered feature expressions to a given domain product features taxonomy using several word similarity metrics. [5] proposed a graph pruning categorization algorithm based on semantic and contextual similarities.

Recently, topic modeling is also used to solve the problem [6, 7]. [6] employed a multi-grain topic model and rating information to identify coherent aspects in the reviews. [7] proposed a MaxEnt-LDA hybrid model to jointly discover both aspects and aspect-specific opinion words. Some studies have also tried to incorporate some prior knowledge into the categorization process [8, 9]. [8] extended a constrained-LDA model with the ability to process large-scale constraints. The most related work [9] transformed the unsupervised feature clustering into a semi-supervised learning problem, which exploited the sharing-words and lexical characteristics of feature expressions to automatically identify some labeled examples.

3   The Proposed Approach

3.1   Constraints Definition and Extraction

In this paper, we mainly define and extract two types of constraints between feature expressions: morphological and contextual constraints, by leveraging the prior observations in domain reviews. According to constraining direction, these constraints are generally divided into two classes: direct and reverse constraints, which indicate the confidence that pair-wise feature expressions should or should not belong to the same categories respectively.

Morphological Constraints: Morphological constraints model the morphological relations between feature expressions. By analyzing the tagged product features, we find that many feature expressions are noun phrases that share some common words, e.g. "wide angle lens" and "lens", "picture quality" and "picture clarity", "price tag" and "price" etc. These feature expressions sharing common words are more likely to belong to the same group. In this paper, we mainly model this sharing words knowledge as direct morphological constraints, which indicate the confidence that pair-wise feature expressions should belong to the same categories.

Morphological constraints are extracted by the morphological analysis between feature expressions. We assume that there exists a morphological constraint between two feature expressions if they share common nouns or noun phrases with the stop words and pronouns being not counted.

Contextual Constraints: Contextual constraints model the contextual relations between product features. In this paper, we mainly focus on reverse contextual constraints that indicate the confidence that pair-wise feature expressions should not belong to the same categories. Contextual constraints are extracted based on following observations:

Feature expressions that ever co-occur in the same sentence are unlikely to belong to the same group, e.g., in sentence "*this camera has an absolutely amazing **zoom**, **optics** are top notch and **macro mode** is incredible*", the feature expressions "***zoom***", "***optics***" and "***macro mode***" are unlikely to belong to the same feature because people are unlikely to repeat the same thing in the same sentence. This intra-sentential co-occurrence knowledge is modeled as reverse contextual constraints.

The feature expressions taking opposite polarities in single review are unlikely to belong to the same aspect because people are unlikely to express contradictory sentiment polarities toward the same feature in single review. This intra-review sentiment consistency knowledge is also modeled as reverse contextual constraints.

We define that at most one class of constraints exists between every pair of feature expressions. When both direct and reverse constraints are conflicting between two feature expressions, both of them are discarded in order to avoid biased constraints.

3.2 Constraint Propagation

After the constraints extraction, we denote the direct constraints collection as $M = \{(x_i, x_j): z_i = z_j\}$ and the reverse constraints collection as $= \{(x_i, x_j): z_i \neq z_j\}$, where z_i is the category label of product feature x_i. To intuitively represent these constraints, we primarily define a constraints matrix $Z_{N \times N}$, where Z_{ij} is defined as the constraints knowledge between x_i and x_j, with $|z_{ij}|$ denotes the confidence:

$$Z_{ij} = \begin{cases} 1, & (x_i, x_j) \in M; \\ -1, & (x_i, x_j) \in R; \\ 0, & otherwise. \end{cases} \quad (1)$$

Since Z only has limited influence on the local pair-wise feature expressions where $|z_{ij}| > 0$, inspired by [10], the constraint propagation is employed to spread the local constraints throughout the whole feature expressions collection.

Let $\bar{E} = \{E = \{E_{ij}\}_{N \times N} : |E_{ij}| \leq 1\}$, where $E \in \bar{E}$ denotes a set of constraints with the associated confidence scores, for $E_{ij} > 0$ is equivalent to $(x_i, x_j) \in M$ while $E_{ij} < 0$ is equivalent to $(x_i, x_j) \in R$, with $|E_{ij}|$ being the confidence score. Given the similarity matrix A between feature expressions, calculate the symmetric weight matrix W for A. The constraint propagation algorithm is defined as follows:

(1). Construct the matrix $L = D^{-1/2} W D^{-1/2}$, where D is a diagonal matrix with its (i, i)-element equals to the sum of the i-th row of W.

(2). Iterate $E_v(t + 1) = \alpha L E_v(t) + (1 - \alpha)Z$ for vertical constraint propagation until convergence, where $E_v(t) \in \bar{E}$ and α is a parameter in the range $(0, 1)$. We empirically set α as 0.5 in this work.

(3). Iterate $E_h(t+1) = \alpha E_h(t)L + (1-\alpha)E_v^*$ for horizontal constraint propagation until convergence, where $E_h(t) \in \bar{E}$ and E_v^* is the limit of $\{E_v(t)\}$.

(4). Output $E^* = E_h^*$ as the final propagated constraints, E_h^* is the limit of $\{E_h(t)\}$.

The above constraint propagation algorithm is proved to have good computation efficiency and convergence ability [10].

3.3   Constrained Spectral Clustering

After the constraint propagation, we get an exhaustive collection of propagated constraints with the associated confidence scores $|E^*|$. Our goal is to obtain a categorization of product features that is consistent with E^*. We exploit E^* for spectral clustering by adjusting the weight matrix W as follows:

$$\tilde{W}_{ij} = \begin{cases} 1 - (1 - E_{ij}^*)(1 - W_{ij}), & E_{ij}^* \geq 0; \\ (1 + E_{ij}^*)W_{ij}, & E_{ij}^* < 0. \end{cases} \tag{2}$$

Here we get a new weight matrix $\tilde{W}$ that incorporated the exhaustive set of propagated constraints obtained by the constraint propagation, then we perform the spectral clustering algorithm with $\tilde{W}$.

4   Experimental Setup

Three product domains of customer reviews: digital camera, cell phone and vacuum cleaner are employed to evaluate our proposed product features categorization strategy. Since this paper only focuses on the product features categorization problem, we assume that feature expressions are already extracted and manually tagged into meaningful categories as the gold standards. The statistics are described in Table l.

Table 1. The statistics of evaluation dataset

Domain	Reviews Num#	Product features Num#	Categories Num#
Digital camera	524	691	35
Cell phone	204	734	40
Vacuum cleaner	856	771	38

The product features are categorized based on their distributional similarities in domain corpus. Each feature expression is represented by the contextual surrounding words in its neighboring windows. The VSM model, tf-idf and Cosine are employed.

We compare the performance of our constrained spectral clustering strategy (CSC) against the traditional K-means clustering (K-means) and spectral clustering (SC) algorithms. K-means clustering partitions the dataset into k clusters in which each instance belongs to the cluster with the nearest mean. The spectral clustering without any constraints is presented to investigate the contribution of our defined constraints.

For the evaluation metrics, we also utilize Entropy and Purity to evaluate the categorization results [9]. Entropy measures the randomness degree of the clustering

results. Purity measures the extent that a category contains only data from one gold-partition.

5   Result Analysis

5.1   Constraints Extraction Results

Table 2. Statistics of extracted constraints for three domains

Domain	Digital camera	Cell phone	Vacuum cleaner
Direct Morphological constraints	9.87%	12.89%	11.35%
Reverse Contextual constraints	8.57%	6.08%	15.19%
Sum of both types of constraints	19.44%	18.97%	26.54%

Table 2 describes the statistics of extracted constraints on three domains respectively. Both the direct morphological constraints and the reverse contextual constraints are counted by their occurrence percentages among all feature expressions. Since morphological constraints are obtained by morphological analysis between feature expressions, they are directly affected by the size of feature expressions, but seldom affected by the size of reviews corpus; while contextual constraints are extracted from sentential and review contexts, they are more sensitive to the size of reviews corpus. It is showed that we achieve the maximum reverse contextual constraints in vacuum cleaner domain due to the largest size of reviews corpus.

5.2   Categorization Results Comparison with Baselines

Table 3. Categorization results comparison with the baselines

Domain	Digital camera		Cell phone		Vacuum cleaner	
Measure	Entropy	Purity	Entropy	Purity	Entropy	purity
K-means	1.648	0.394	1.428	0.497	1.823	0.390
SC	1.875	0.312	1.406	0.454	1.834	0.411
CSC	1.574	0.457	1.321	0.528	1.603	0.514

Table 3 describes results comparison with the baselines on three domains respectively. It is showed that our proposed CSC method always achieves the best entropy and purity performance. Compared with the basic K-means clustering, CSC achieves obvious better performance. Compared with the spectral clustering without any constraints, CSC also achieves obvious improvement, which verifies the contribution of the morphological and contextual constraints to product features categorization.

6   Conclusion and Future Work

In this paper, we propose a semi-supervised product features categorization strategy using constrained spectral clustering. The morphological and contextual characteristics

of this problem are modeled as constraints, and exploited as prior knowledge to improve the categorization performance. The local constraints are spread by the constraint propagation, and incorporated into spectral product features clustering globally. Empirical evaluation on real-life dataset has demonstrated the effectiveness of our proposed strategy compared with the state-of-art baselines.

Since the product features do not always exhibit as a flat structure that can be partitioned clearly, our future work will devote to cluster them into a fine-grained hierarchical structure.

References

1. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of the 14th International Conference on World Wide Web, WWW 2005, pp. 342–351 (2005)
2. Guo, H., Zhu, H., Guo, Z., et al.: Product feature categorization with multilevel latent semantic association. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, pp. 1087–1096 (2009)
3. Su, Q., Xu, X., Guo, H., et al.: Hidden sentiment association in chinese web opinion mining. In: Proceedings of the 17th International Conference on World Wide Web, WWW 2008, pp. 959–968 (2008)
4. Carenini, G., Ng, R.T., Zwart, E.: Extracting knowledge from evaluative text. In: Proceedings of the 3rd International Conference on Knowledge Capture, K-CAP 2005, pp. 11–18 (2005)
5. Huang, S., Liu, X., Peng, X., et al.: Fine-grained product features extraction and categorization in reviews opinion mining. In: Proceedings of 2012 IEEE 12th International Conference on Data Mining Workshops, ICDM 2012, pp. 680–686 (2012)
6. Titov, I., McDonald, R.: Modeling online reviews with multi-grain topic models. In: Proceedings of the 17th International Conference on World Wide Web, WWW 2008, pp. 111–120 (2008)
7. Zhao, W.X., Jiang, J., Yan, H., Li, X.: Jointly modeling aspects and opinions with a Max-Ent-LDA hybrid. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, pp. 56–65 (2010)
8. Zhai, Z., Liu, B., Xu, H., Jia, P.: Constrained LDA for grouping product features in opinion mining. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part I. LNCS, vol. 6634, pp. 448–459. Springer, Heidelberg (2011)
9. Zhai, Z., Liu, B., Xu, H., Jia, P.: Clustering product features for opinion mining. In: Proceedings of the fourth ACM International Conference on Web Search and Data Mining, WSDM 2011, pp. 347-354 (2011)
10. Lu, Z., Ip, H.H.S.: Constrained spectral clustering via exhaustive and efficient constraint propagation. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 1–14. Springer, Heidelberg (2010)

A New Approach for Improving Cross-Document Knowledge Discovery Using Wikipedia

Peng Yan and Wei Jin

Department of Computer Science, North Dakota State University
1340 Administration Ave., Fargo, ND 58102, USA
{peng.yan,wei.jin}@ndsu.edu

Abstract. In this paper, we present a new model that incorporates the extensive knowledge derived from Wikipedia for cross-document knowledge discovery. The model proposed here is based on our previously introduced Concept Chain Queries (CCQ) which is a special case of text mining focusing on detecting semantic relationships between two concepts across multiple documents. We attempt to overcome the limitations of CCQ by building a semantic kernel for concept closeness computing to complement existing knowledge in text corpus. The experimental evaluation demonstrates that the kernel-based approach outperforms in ranking important chains retrieved in the search results.

Keywords: Knowledge Discovery, Semantic Relatedness, Cross-Document knowledge Discovery, Document Representation.

1 Introduction

Traditionally text documents are represented as a Bag of Words (BOW) and the semantic relatedness between concepts are measured based on statistical information from the corpus such as the widely used tf-idf weighting scheme [3], [7]. The main theme of our previous introduced Concept Chain Queries (CCQ) [3] was specifically designed to discover semantic relationships between two concepts across documents where relationships found reveal semantic paths linking two concepts across multiple text units. However, only the BOW model was used in CCQ for text representation and thus the techniques proposed in [3] have the inborn limitations. For example, Ziyad Khaleel, also known as Khalil Ziyad was a Palestinian-American al-Qaeda member, based in the United States, being identified as a "procurement agent" for Bin Ladin's terroristic organization. Clearly he has a close relationship with Bin Ladin. Nevertheless, he will not be taken into consideration if his name does not appear in the document collection where the concept chain queries are performed. To alleviate such limitations, this effort proposes a new model that has a semantic kernel built inside to embed the extensive knowledge from Wikipedia into the original knowledge base, aiming at taking advantage of outside knowledge to improve cross-document knowledge discovery. Here we employ the Explicit Semantic Analysis (ESA) technique introduced by Gabrilovich et al. [1] to help build an ESA-based kernel that captures the semantic closeness of concepts in a much larger knowledge space.

Our contribution of this effort can be summarized as follows. First, in comparison to traditional methods mostly based on the BOW representation, the proposed model is able to provide a much more comprehensive knowledge repository to support various queries and effectively complements existing knowledge contained in text corpus. Second, built on the traditional BOW text representation for content analysis, we successfully integrate ESA into the process of knowledge discovery to help measure the semantic relatedness between concepts. We envision this integration would also benefit other related tasks such as question answering and cross–document summarization. Third, we build an ESA-based kernel that is capable of measuring semantic relatedness between concepts by considering the comprehensive knowledge derived from Wikipedia. It would be also convenient to re-use this kernel in other research fields that involve semantic relevance computing. Last, the model proposed in this work generates ranked concept chains where the key terms representing significant relationships between topics are ranked higher.

2   Related Work

There have been a great number of text mining algorithms for capturing relationships between concepts developed [3], [5], [6], [7]. However, built on the traditional Bag-of-Words (BOW) representation with no or little background knowledge being taken into account, those efforts achieved a limited discovery scope. Hotho et al. [2] exploited WordNet to improve the BOW text representation and Martin [4] developed a method for transforming the noun-related portions of WordNet into a lexical ontology to enhance knowledge representation. These techniques suffer from relatively limited coverage and painful maintenance of WordNet compared to Wikipedia, the world's largest knowledge base to date. [8] embeds background knowledge derived from Wikipedia into a semantic kernel to enrich document representation for text classification. The empirical evaluation demonstrates their approach successfully achieves improved classification accuracy. However, their method is based on a thesaurus built from Wikipedia and constructing the thesaurus requires a considerable amount of effort. Our proposed solution is motivated by [1], [8], and to tackle the above problems, we 1) adapt the ESA technique to better suit our task and further develop a sequence of heuristic strategies to filter out irrelevant terms and retain only top-k most relevant concepts to the given topics; 2) build an ESA-based kernel which requires much less computational effort to measure the closeness between concepts using Wiki knowledge.

3   Kernel Method

We adapt the Explicit Semantic Analysis (ESA) to remove noise concepts derived from Wikipedia [9], and then build a semantic kernel for semantic relatedness computing. The basic idea of kernel methods is to embed the data in a suitable feature space (with more information integrated), such that solving the problem in the new space is easier (e.g. linear). To be exact, the new space here stands for the space that incorporates

Wikipedia knowledge, and the kernel represents the semantic relationship between two concepts/topics uncovered in this new space.

3.1   Building the ESA-Based Kernel

The purpose of building the ESA-based kernel is in concern of word semantics omission in the BOW model where feature weight is calculated only considering the number of occurrences. To build the semantic kernel for a given topic, we first need to transform the concept vector constructed using the BOW model into a different vector (i.e. space transformation) with new knowledge embedded. Suppose the topic T is represented by a weighted vector of concepts: $\phi(T) =< c_1, c_2, ..., c_n >$ using the BOW model. The value of each element in the vector corresponds to a tf-idf value. We then define a kernel matrix M for the topic T as show in Table 1.

Table 1. The kernel matrix

	c_1	c_2	$\cdots$	c_n
c_1	1	x	$\cdots$	y
c_2	x	1	$\cdots$	z
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_n	y	z	$\cdots$	1

M is a symmetrical matrix and the elements fell on the diagonal line are all equal to 1, since according to ESA, the same concept has the same interpretation vector, which means the ESA-based similarity between two same concepts is 1. Formally, M is defined as below:

$$M_{i,j} = \begin{cases} 1 & if \quad i = j \\ Sim_{ESA}(c_i, c_j) / Sim\_Max & if \quad i \neq j \end{cases} \tag{1}$$

Where $Sim_{ESA}(c_i, c_j)$ is the ESA similarity between c_i and c_j, and Sim_Max is the maximum value in M besides the elements on the diagonal line. Then a transformation of $\phi(T)$ can be achieved through: $\tilde{\phi}(T) = \phi(T)M$, where $\tilde{\phi}(T)$ represents the topic T in a linear space with much more information integrated. With $\tilde{\phi}(T)$, the ESA-based kernel between two topics T_1 and T_2 can be represented as:

$$k(T_1, T_2) = \phi(T_1)MM^T\phi(T_2)^T$$
$$= \phi(T_1)M(\phi(T_2)M)^T \tag{2}$$
$$= \tilde{\phi}(T_1)\tilde{\phi}(T_2)^T$$

Therefore, the semantic relationship between two topics is now represented using the ESA-based kernel i.e. $k(T_1, T_2)$ which incorporates Wiki knowledge.

3.2   Improving Semantic Relatedness Computing

Once the relevant concepts for a topic of interest have been identified using CCQ [3], we are ready to use ESA-based kernel to help compute the semantic relatedness between concepts. For example, given the concept "*Clinton*" as a topic of interest, and the BOW-based concept vector for "*Clinton*" and the corresponding kernel matrix are shown in Table 2 and Table 3. Table 4 illustrates the improvement through multiplying the BOW-based concept vector by the kernel matrix. This is consistent with our understanding that Hillary as Clinton's wife should be considered most related to him. Shelton, served as Chairman of the Joint Chiefs of Staff during Clinton's terms in office, stays in the second position. At last, Clancy, who hardly has a relationship with Clinton is degraded to the end of the vector.

Table 2. The BOW-based concept vector for "Clinton"

	Clancy	Shelton	Hillary
Clinton	0.54	0.43	0.38

Table 3. The kernel matrix for "Clinton"

	Clancy	Shelton	Hillary
Clancy	1	0.113	0.147
Shelton	0.113	1	1
Hillary	0.147	1	1

Table 4. The improved concept vector for "Clinton"

	Hillary	Shelton	Clancy
Clinton	0.889	0.871	0.644

We apply the ESA-based kernel to CCQ in the following steps:

1. Conduct independent searches for A and C. Build the A and C profiles. Call these profiles AP and CP respectively.
2. Compute a B profile (BP) composed of terms in common between AP and CP. The corpus-level weight of a concept in BP is the sum of its weights in AP and CP. This is the first level of intermediate potential concepts generated from the text corpus.
3. Build the kernel matrix for all the concepts in BP, and update the weight of each concept in BP using the kernel matrix.
4. Expand the concept chains using the created BP profile together with the topics to build additional levels of intermediate concept lists DP and EP which (i) connect the topics to each concept in BP profile in the sentence level within each semantic type, and (ii) also normalize and rank them.
5. Build the kernel matrix for DP and EP respectively by following the same way in Step 3, and then update the weight of each concept in DP and EP.

4   Empirical Evaluation

4.1   Evaluation Data

An open source document collection pertaining to the 9/11 attack, including the publicly available 9/11 commission report was used in our evaluation. The report consists of Executive Summary, Preface, 13 chapters, Appendix and Notes. Each of them was considered as a separate document resulting in 337 documents. Query pairs selected by the assessors covering various scenarios (e.g., ranging from popular entities to rare entities) were conducted and used as our evaluation data.

4.2   Experimental Results

Table 5 through 8 summarize the results we obtain on executing concept chain queries from the evaluation set. Table 5 and 6 measure how often the truth chain was generated as we kept the top 5 and top 10 concepts within each semantic type respectively. We observe that the search performance has been significantly improved by apply the ESA-based kernel. Table 7 and 8 show the improvement of average rank of the concepts retrieved in the search results. It is demonstrated that the kernel-based approach outperforms a lot over the BOW-based approach in ranking key terms representing significant relationships between topics.

Table 5. Search results for keeping top 5 concepts

Model	No. of Top Rank			
	Length 1	Length 2	Length 3	Length 4
BOW-based Approach	10/14	23/42	15/28	5/23
Kernel-based Approach	10/14	33/42	21/28	11/23

Table 6. Search results for keeping top 10 concepts

Model	No. of Top Rank			
	Length 1	Length 2	Length 3	Length 4
BOW-based Approach	10/14	30/42	22/28	13/23
Kernel-based Approach	10/14	38/42	22/28	17/23

Table 7. Average rank for keeping top 5 concepts

Model	Average Rank			
	Length 1	Length 2	Length 3	Length 4
BOW-based Approach	1	2.74	3.13	5.20
Kernel-based Approach	1	1.82	2.71	4.00

Table 8. Average rank for keeping top 10 concepts

Model	Average Rank			
	Length 1	Length 2	Length 3	Length 4
BOW-based Approach	1	11.15	4.73	9.23
Kernel-based Approach	1	3.05	2.91	7.35

5   Conclusion and Future Work

A Wikipedia-integrated mining model is presented to overcome the limitations of the Bag-of-Words approach. We propose a method to construct a light-weight kernel using Explicit Semantic Analysis for measuring the semantic closeness between concepts. The semantic kernel is applied to the process of generating multiple levels of concept chains. By incorporating the knowledge derived from Wikipedia, this methodology is able to boost concepts that are most closely related to the topics to higher rankings.

References

1. Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In: 20th International Joint Conference on Artificial Intelligence, pp. 1606–1611. Morgan Kaufmann, San Francisco (2007)
2. Hotho, A., Staab, S., Stumme, G.: Wordnet improves Text Document Clustering. In: SIGIR 2003 Semantic Web Workshop, pp. 541–544. Citeseer (2003)
3. Jin, W., Srihari, R.: Knowledge Discovery across Documents through Concept Chain Queries. In: 6th IEEE International Conference on Data Mining Workshops, pp. 448–452. IEEE Computer Society, Washington (2006)
4. Martin, P.A.: Correction and Extension of WordNet 1.7. In: de Moor, A., Ganter, B., Lex, W. (eds.) ICCS 2003. LNCS (LNAI), vol. 2746, pp. 160–173. Springer, Heidelberg (2003)
5. Srinivasan, P.: Text Mining: Generating hypotheses from Medline. Journal of the American Society for Information Science and Technology 55(5), 396–413 (2004)
6. Srihari, R.K., Lamkhede, S., Bhasin, A.: Unapparent Information Revelation: A Concept Chain Graph Approach. In: 14th ACM International Conference on Information and Knowledge Management, pp. 329–330. ACM, New York (2005)
7. Swason, D.R., Smalheiser, N.R.: Implicit Text Linkage between Medline Records: Using Arrowsmith as an Aid to Scientific Discovery. Library Trends 48(1), 48–59 (1999)
8. Wang, P., Domeniconi, C.: Building Semantic Kernels for Text Classification using Wikipedia. In: 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 713–721. ACM, New York (2008)
9. Yan, P., Jin, W.: Improving Cross-Document Knowledge Discovery Using Explicit Semantic Analysis. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2012. LNCS, vol. 7448, pp. 378–389. Springer, Heidelberg (2012)

Using Grammar-Profiles to Intrinsically Expose Plagiarism in Text Documents

Michael Tschuggnall and Günther Specht

Databases and Information Systems
Institute of Computer Science, University of Innsbruck
{michael.tschuggnall,guenther.specht}@uibk.ac.at

Abstract. Intrinsic plagiarism detection deals with the task of finding plagiarized sections in text documents without using a reference corpus. This paper describes a novel approach in this field by analyzing the grammar of authors and using sliding windows to find significant differences in writing styles. To find suspicious text passages, the algorithm splits a document into single sentences, calculates syntax grammar trees and builds profiles based on frequently used grammar patterns. The text is then traversed, where each window is compared to the document profile using a distance metric. Finally, all sentences that have a significantly higher distance according to a utilized Gaussian normal distribution are marked as suspicious. A preliminary evaluation of the algorithm shows very promising results.

Keywords: intrinsic plagiarism detection, pq-gram profiles, grammar trees, stylistic inconsistencies, NLP applications.

1 Introduction

The huge amount of publicly available text documents makes it increasingly easier for authors to copy suitable text fragments into their works. On the other side, the task of identifying plagiarized passages becomes increasingly more difficult for software algorithms that have to deal with large amounts of possible sources. An even harder challenge is to find plagiarism in text documents where the majority of sources is composed of books and other literature that is not digitally available. Nevertheless, more and more recent events show that especially in such cases it would be important to have reliable tools that indicate possible misuses.

The two main approaches for detecting plagiarism in text documents are *external* and *intrinsic* methods, respectively. Given a suspicious document, external algorithms compare text fragments with any available sources (e.g. collections from the world wide web), whereas intrinsic algorithms try to detect plagiarism by inspecting the suspicious document only. Frequently applied techniques in both areas as well as in related topics such as authorship identification or text categorization include n-gram comparisons or standard IR techniques like common subsequences [2] combined with machine learning techniques [3].

The idea of the approach described in this paper is to use a syntactical feature, namely the grammar used by an author, to identify passages that might have been plagiarized. Due to the fact that an author has many different choices of how to formulate a sentence using the existing grammar rules of a natural language, the assumption is that the way of constructing sentences is significantly different for individual authors. For example, the famous Shakespeare quote *"To be, or not to be: that is the question."* (1) could also be formulated as *"The question is whether to be or not to be."* (2) or even *"The question is whether to be or not."* (3) which is semantically equivalent but differs significantly according to the syntax. The main idea of this approach is to quantify those differences by creating a grammar profile of a document and to utilize sliding windows techniques to find suspicious text sections.

The rest of this paper is organized as follows: Section 2 describes the algorithm in detail, while a preliminary evaluation of it is shown in Section 3. Finally, Section 4 sketches related work and Section 5 summarizes the main ideas and discusses future work.

2   The PQ-PlagInn Algorithm

The PQ-PlagInn is a variation of the *PlagInn*[1] algorithm proposed in [14], which calculates grammar trees of all sentences of a document to analyze their building structure in order to find *outstanding* sentences. Instead of calculating a distance matrix using tree edit distances, in this approach we build syntax profiles of the whole document and compare text sections with the document profile using sliding windows. More concretely, the algorithm consists of the following steps:

1. At first the document is cleaned to contain alphanumeric characters and punctuation marks only. Then it is parsed and split into single sentences, which is currently implemented with the open source tool *OpenNLP*[2].
2. Each sentence is then anaylzed by its grammar, i.e. a full syntax grammar tree is calculated using the *Stanford Parser* [7]. For example, Figure 1 depicts the grammar trees resulting from analyzing sentences (1), (2) and (3). The labels of each tree correspond to a Penn Treebank tag [9], where e.g *NP* corresponds to a noun phrase or *JJS* corresponds to a superlative adjective. In order to examine the building structure of sentences only, the concrete words, i.e. the leafs of the tree, are ignored.
3. Having computed a grammar tree for every sentence, the pq-gram index [1] of each tree is calculated in the next step. Pq-grams consist of a stem (p) and a base (q) and can be related to as "n-grams for trees". Thereby p defines how much nodes are included vertically, and q defines the number of nodes to be considered horizontally. For example, a valid pq-gram with $p = 2$ and $q = 3$ starting from level two of tree (1) shown in Figure 1 would be `[S-VP-VP-CC-RB]`. In order to obtain all pq-grams, the base is

---

[1] PlagInn stands for *In*trinsic *Plag*iarism detection *Inn*sbruck.
[2] `http://incubator.apache.org/opennlp`, visited February 2013.
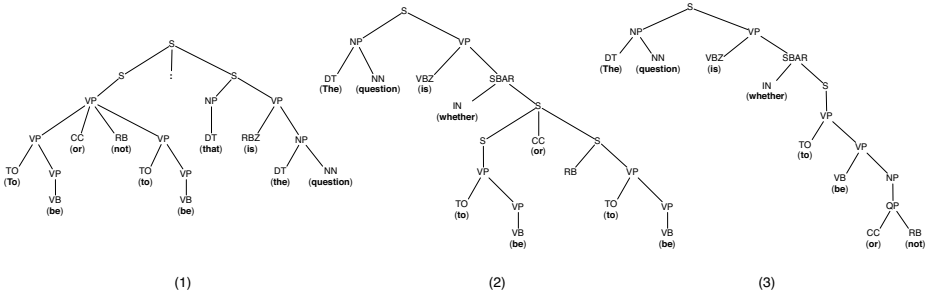
Fig. 1. Grammar Trees Resulting From Parsing Sentence (1), (2) and (3)

shifted left and right additionally: If then less than p nodes exist horizon-
tally, the corresponding pq-gram is filled with $*$ for missing nodes. Therefore
also the pq-grams [S-VP-*-*-VP], [S-VP-*-VP-CC], [S-VP-RB-VP-*] or
[S-VP-VP-*-*] are valid. Finally, the pq-gram index contains all valid pq-
grams of a grammar tree, whereby multiple occurences of the same pq-grams
are also present multiple times in the index.

4. Subsequently, the pq-gram profile of the whole document is calculated by
   combining all pq-gram indexes of all sentences. In this step the number of
   occurences is counted for each pq-gram and then normalized by the doc-
   ument length, i.e. normalized by the total number of distinct pq-grams.
   As an example, the three mostly used pq-grams of a selected document
   are: {[NP-NN-*-*-*], 2.7%}, {[PP-IN-*-*-*], 2.3%}, {[S-VP-*-*-VBD],
   1.1%}. The *pq-gram profile* then consists of the complete table of pq-grams
   and their occurences in the given document, indicating the *favours* or the
   style of syntax construction used by the (main) author.

5. The basic idea is now to utilize sliding windows and calculate the distance for
   each window compared to the pq-gram profile. A window has a predefined
   length l which defines how many sentences should be contained, and the
   window step s defines the starting points of the windows.
   Then for each window the pq-gram profile $P(w)$ is calculated and compared
   to the pq-gram profile of the whole document. For calculating the distance,
   the measure proposed in [12] has been used, as it is well suited for com-
   paring short text fragments (the window w) with large text fragments (the
   document D):

$$d(w, D) = \sum_{p \in P(w)} \left(\frac{2(f_w(p) - f_D(p))}{f_w(p) + f_D(p)} \right)^2$$

Thereby $f_w(p)$ and $f_D(p)$ denote the normalized frequencies of occurences of
pq-gram p in the window w and the document D, respectively. An example
of the sliding window distances of a whole document is illustrated in Figure
2. It shows the distance for each sliding window starting position, using a
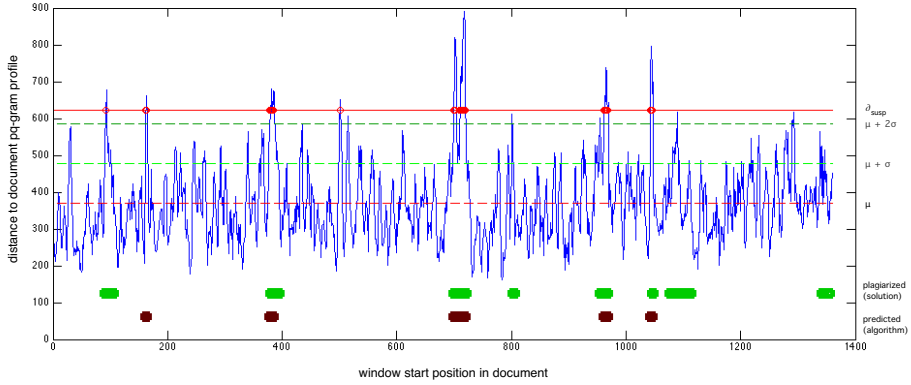window length of $l = 5$ and a window step of $s = 1$ in this case.

Fig. 2. Distances of pq-gram Occurences of Sliding Windows Compared to the Document Profile

6. As it can already be seen visually, some sliding windows differ significantly more than others, which may be the case because they have been plagiarized. The final decision of whether sentences should be predicted to be plagiarized or not is made similar to the method used in the original PlagInn-algorithm by fitting a gaussian normal distribution function and estimating the mean μ and standard deviation σ. Subsequently, all sentences having a higher threshold than δ_{susp} (where $\delta_{susp} \gg \mu+\sigma$) are marked as suspicious. Moreover, the sentence selection algorithm proposed in [14] is applied to either filter out standalone suspicious sentences or to add non-suspicious sentences to groups of suspicious sentences if they reside in between. The bottom of the diagram shown in Figure 2 depicts the final prediction of the algorithm together with the correct solution obtained from the corresponding test set annotations of the document.

3   Evaluation

The PQ-PlagInn algorithm has been preliminary evaluated by using 50 random documents of the PAN 2011 test corpora [11]. All documents are written in English, and the set has been chosen to be heterogenously distributed, i.e. containing short and large documents, with and without pagiarism. The results shown in Figure 3 indicate that this approach is very well suited for the task of intrinsic plagiarism detection. By varying the window length and window step a promising[3] F-score of about 44% could be reached using $l = 8$, $s = 2$ and the pq-gram configuration $p = 2$ and $q = 3$. Thus, the PQ-PlagInn variation indicates to outperform the original PlagInn approach which achieved 35% over the whole

---

[3] Compared to current approaches, e.g. [12] which achieves an F-score of 33%.

test corpus (over 4000 documents). The F-scores are composed of high precision values compared to low recall values. For example, the best performance resulted from a precision value of about 75% and a recall value of about 31%, indicating that if the algorithm predicts a text passage to be plagiarized it is often correct, but on the other hand lacks of finding all passages.
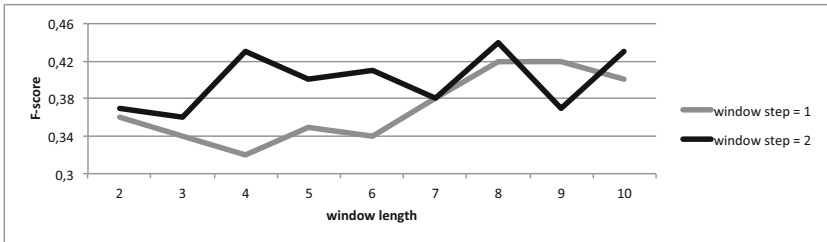

Fig. 3. Evaluation Results

4 Related Work

An often applied concept in the field of intrinsic plagiarism detection is the usage of n-grams [12,6], where the document is split up into chunks of three or four letters, grouped and - as proposed with the algorithm in this paper - analyzed through sliding windows. Another approach also uses the sliding window technique but is based on word frequencies, i.e. the assumption that the set of words used by authors is significantly different [10]. Approaches in the field of author detection and genre categorization also use NLP tools to analyze documents based on syntactic annotations [13]. Word- and text-based statistics like the average sentence length or the average parse tree depth are used in [5].

Another interesting approach used in authorship attribution that tries to detect the writing style of authors by analyzing the occurences and variations of spelling errors is proposed in [8]. It is based on the assumption that authors tend to make similar spelling and/or grammar errors and therefore uses this information to attribute authors to unseen text documents.

Lexicalized tree-adjoining-grammars (LTAG) are poposed in [4] as a ruleset to construct and analyze grammar syntax by using partial subtrees, which may also be used with this approach as an alternative to pq-gram patterns.

5 Conclusion and Future Work

In this paper a new approach for intrinsic plagiarism detection is presented which tries to find suspicious sentences by analyzing the grammar of an author. It builds a grammar-profile using pq-grams of syntax trees of a suspicious text document and compares it by utilizing sliding windows. A preliminary evaluation using 50 random documents reached a promising F-score of about 44%. Manual

inspections showed that the algorithm produces high precision values, i.e. mostly predicts plagiarism only where this is really the case. On the other hand it could be improved to find more plagiarism cases, i.e. increasing the recall value.

Future work should also evaluate the approach against a larger and more representative test set. Additionally, all parameters like window length, window step, pq-gram configurations or other thresholds should be optimized. As currently no lexical information is used, a combination with existing approaches could enhance the overall performance as well as the adaption to more (syntactically complex) languages. Finally, the PQ-PlagInn algorithm also seems to be very suitable for tasks in the field of authorship attribution/verification or text categorization, and it should thus be adjusted and accordingly evaluated.

References

1. Augsten, N., Böhlen, M., Gamper, J.: The pq-Gram Distance between Ordered Labeled Trees. ACM Transactions on Database Systems, TODS (2010)
2. Gottron, T.: External Plagiarism Detection Based on Standard IR Technology and Fast Recognition of Common Subsequences. In: CLEF (Notebook Papers/LABs/Workshops) (2010)
3. Joachims, T.: Text Categorization with Suport Vector Machines: Learning with Many Relevant Features. In: Proceedings of the 10th European Conference on Machine Learning, London, UK, pp. 137–142 (1998)
4. Joshi, A.K., Schabes, Y.: Tree-Adjoining Grammars. Handbook of Formal Languages 3, 69–124 (1997)
5. Karlgren, J.: Stylistic Experiments For Information Retrieval. PhD thesis, Swedish Institute for Computer Science (2000)
6. Kestemont, M., et al.: Intrinsic Plagiarism Detection Using Character Trigram Distance Scores. In: CLEF Labs and Worksh. Papers, Amsterdam, Netherlands (2011)
7. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proc. of the 41st Meeting on Comp. Linguistics, Stroudsburg, PA, USA, pp. 423–430 (2003)
8. Koppel, M., Schler, J.: Exploiting Stylistic Idiosyncrasies for Authorship Attribution. In: IJCAI 2003 Workshop on Computational Approaches to Style Analysis and Synthesis, pp. 69–72 (2003)
9. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: The Penn Treebank. Comp. Linguistics 19, 313–330 (1993)
10. Oberreuter, G., et al.: Approaches for Intrinsic and External Plagiarism Detection. In: Notebook Papers of CLEF Labs and Workshops (2011)
11. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An Evaluation Framework for Plagiarism Detection. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Beijing, China (2010)
12. Stamatatos, E.: Intrinsic Plagiarism Detection Using Character n-gram Profiles. In: CLEF (Notebook Papers/Labs/Workshop) (2009)
13. Stamatatos, E., Kokkinakis, G., Fakotakis, N.: Automatic text categorization in terms of genre and author. Comput. Linguist. 26, 471–495 (2000)
14. Tschuggnall, M., Specht, G.: Detecting Plagiarism in Text Documents through Grammar-Analysis of Authors. In: 15. GI-Fachtagung Datenbanksysteme für Business, Technologie und Web, Magdeburg, Germany (2013)

Feature Selection Methods in Persian Sentiment Analysis

Mohamad Saraee[1] and Ayoub Bagheri[2]

[1] School of Computing, Science and Engineering, University of Salford, Manchester, UK
m.saraee@salford.ac.uk
[2] Intelligent Database, Data Mining and Bioinformatics Lab,
Electrical and Computer Engineering Dep., Isfahan University of Technology, Isfahan, Iran
a.bagheri@ec.iut.ac.ir

Abstract. With the enormous growth of digital content in internet, various types of online reviews such as product and movie reviews present a wealth of subjective information that can be very helpful for potential users. Sentiment analysis aims to use automated tools to detect subjective information from reviews. Up to now as there are few researches conducted on feature selection in sentiment analysis, there are very rare works for Persian sentiment analysis. This paper considers the problem of sentiment classification using different feature selection methods for online customer reviews in Persian language. Three of the challenges of Persian text are using of a wide variety of declensional suffixes, different word spacing and many informal or colloquial words. In this paper we study these challenges by proposing a model for sentiment classification of Persian review documents. The proposed model is based on stemming and feature selection and is employed Naive Bayes algorithm for classification. We evaluate the performance of the model on a collection of cellphone reviews, where the results show the effectiveness of the proposed approaches.

Keywords: sentiment classification, sentiment analysis, Persian language, Naive Bayes algorithm, feature selection, mutual information.

1 Introduction

In the recent decade, with the enormous growth of digital content in internet and databases, sentiment analysis has received more and more attention between information retrieval and natural language processing researchers. Up to now, many researches have been conducted sentiment analysis on English, Chinese or Russian languages [1-9]. However on Persian text, in our knowledge there is little investigation conducted on sentiment analysis [10]. Persian is an Indo-European language, spoken and written primarily in Iran, Afghanistan, and a part of Tajikistan. The amount of information in Persian language on the internet has increased in different forms. As the style of writing in Persian language is not firmly defined on the web, there are too many web pages in Persian with completely different writing styles for the same words [11, 12]. Therefore in this paper, we study a model of feature selection in sentiment classification for Persian language, and experiment our model on a Persian

product review dataset. In the reminder of this paper, Section 2 describes the proposed model for sentiment classification of Persian reviews. In Section 3 we discuss important experimental results, and finally we conclude with a summary in section 5.

2   Proposed Model for Persian Sentiment Analysis

Persian sentiment analysis suffers from low quality, where the main challenges are
- Lack of comprehensive solutions or tools
- Using of a wide variety of declensional suffixes
- Word spacing
  - In Persian in addition to white space as inter-words space, an intra-word space called pseudo-space separates word's part.
- Utilizing many informal or colloquial words

In this paper, we propose a model, using n-gram features, stemming and feature selection to overcome the Persian language challenges in sentiment classification.

2.1   Sentiment Classifier

In this paper, we consider Naive Bayes algorithm which is a machine learning approach as the sentiment classifier [13]. In the problem of sentiment classification we use vector model to represent the feature space. For the feature space we extract n-gram features to deal with the conflicting problem of space and pseudo-space in Persian sentences. Here we use unigram and bigram phrases as n-gram features. Therefore in this model, the sequence of the words is important. Experiments show using n-gram features could solve the problem of different word spacing in Persian text.

2.2   Feature Selection for Sentiment Analysis

Feature Selection methods sort features on the basis of a numerical measure computed from the documents in the dataset collection, and select a subset of the features by thresholding that measure. In this paper four different information measures were implemented and tested for feature selection problem in sentiment analysis. The measures are Document Frequency (DF), Term Frequency Variance (TFV), Mutual Information (MI) [14] and Modified Mutual Information (MMI). Below we discuss presented MMI approach.

Mutual Information and Modified Mutual Information
In this paper we introduce a new approach for feature selection, Modified Mutual Information. In order to explain MMI measure, it is helpful to first introduce Mutual Information by defining a contingency table (see Table 1).

Table 1. Contingency table for features and classes

	c	$\bar{c}$
f	A	B
$\bar{\bar{f}}$	C	D

Table 1 records co-occurrence statistics for features and classes. We also have that the number of review documents, N = A+B +C +D. These statistics are very useful for estimating probability values [13, 14]. By using Table 1, MI can be computed by equation (7):

$$MI(f,c) = \log\frac{P(f,c)}{P(f)P(c)} \tag{1}$$

Where $P(f,c)$ is the probability of co-occurrence of feature f and class c together, and $P(f)$ and $P(c)$ are the probability of co-occurrence of feature f and class c in the review documents respectively. Therefore by Table 1, MI can be approximated by Equation (8):

$$MI(f,c) = \log\frac{A*N}{(A+B)*(A+C)} \tag{2}$$

Intuitively MI measures if the co-occurrence of f and c is more likely than their independent occurrences, but it doesn't measure the co-occurrence of f and $\bar{c}$ or the co-occurrence of other features and class c. We introduce a Modified version of Mutual Information as MMI which consider all possible combinations of co-occurrences of a feature and class label. First we define four parameters as the following:

- $p(f,c)$: *Probability of co-occurrence of feature f and class c together.*
- $p(\bar{f},\bar{c})$: *Probability of co-occurrence of all features except f in all classes except c together.*
- $p(\bar{f},c)$: *Probability of co-occurrence of all features except feature f in class c.*
- $p(f,\bar{c})$: *Probability of co-occurrence of feature f in all classes except c.*

We calculate MMI score as Equation (10):

$$MMI(f,c) = \log\frac{p(f,c)*p(\bar{f},\bar{c})}{p(f)*p(c)*p(\bar{f})*p(\bar{c})} - \log\frac{p(\bar{f},c)*p(f,\bar{c})}{p(f)*p(c)*p(\bar{f})*p(\bar{c})} \tag{3}$$

Where $P(f)$ and $P(c)$ are the probability of independent occurrence of feature f and class c in the review documents respectively. $p(\bar{f})$ is the number of review documents which not contain feature f and $p(\bar{c})$ is the number of documents with the classes other than class c. Based on Table 4, MMI can be approximated by Equation:

$$MMI(f,c) = \frac{A*D - C*B}{(A+C)*(B+D)*(A+B)*(C+D)} \tag{4}$$

3   Experimental Results

To test our methods we compiled a dataset of 829 online customer reviews in Persian language from different brands of cell phone products. We assigned two annotators to label customer reviews by selecting a positive or negative polarity on the review level. After annotation, the dataset reached to 511 positive and 318 negative reviews.

3.1   Comparative Study

In our experiments, first we evaluated Persian sentiment classification in two phases:

Phase 1. *Without n-gram features and stemming*
Phase 2. *With n-gram features and stemming*

Table 2 shows the F-score results for the two phases. From the results we can observe that using of n-gram features and stemming for sentiment classification has 4% and 0.3% improvements for negative and positive classes respectively.

Table 2. F-scores for phases 1 and 2, Without and with n-gram features and stemming

Phase	Class	F-score
1	Negative	0.7480
	Positive	0.8570
2	Negative	0.7880
	Positive	0.8600

In this work we applied four different feature selection approaches, MI, DF, TFV and MMI with the Naive Bayes learning algorithm to the online Persian cellphone reviews. In the experiments, we found that using feature selection with learning algorithms can perform improvement to classifications of sentiment polarities of reviews.

Table 3 indicates Precision, Recall and F-score measures on two classes of Positive and Negative polarity with the feature selection approaches.

Table 3. Precision, Recall and F-score measures for the feature selection approaches with naive bayes classifier

Approach	Class	Precision	Recall	F-score
MI	Negative	0.4738	0.8356	0.6026
	Positive	0.8130	0.4260	0.5538
DF	Negative	0.8148	0.7812	0.7962
	Positive	0.8692	0.8898	0.8788
TFV	Negative	0.8226	0.7800	0.7996
	Positive	0.8680	0.8956	0.8814
MMI	Negative	0.7842	0.8568	0.8172
(Proposed Approach)	Positive	0.9072	0.8526	0.8784

The results from Table 3 indicate that the TFV, DF and MMI have better performances than the traditional MI approach. In terms of F-score, MMI improves MI with 21.46% and 32.46% on Negative and Positive classes respectively, DF overcomes MI with 19.36% and 32.5% better performances for Negative and Positive review documents respectively and TFV improves MI with19.7% and 32.76% for Negative and Positive documents respectively. The reason of poor performance for MI is that of MI only uses the information between the corresponding feature and the corresponding class and does not utilize other information about other features and other classes. When we compare DF, TFV and MMI, we can find that the MMI beats both DF and TFV on F-scores of Negative review documents with 2.1% and 1.76% improvements respectively, but for the Positive review documents DF and TFV have 0.04% and 0.3% better performance than the MMI, respectively.

To assess the overall performance of techniques we adopt the macro and micro average, Figure 1 shows the macro and micro average F-score.
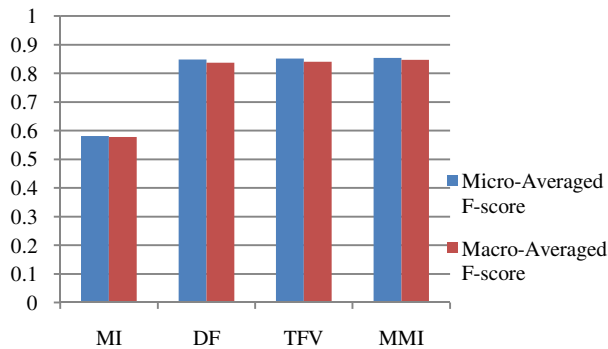

Fig. 1. Macro and micro average F-score for MI, DF, TFV and MMI

From this Figure we can find that the MMI proposed approach has slightly better performance than the DF and TFV approaches and has significant improvements on MI method. The basic advantage of the MMI is using of whole information about a feature, positive and negative factors between features and classes. MMI in overall can reach to 85% of F-score classification. It is worth noting that with a larger training corpus the feature selection approaches and the learning algorithm could get higher performance values. Additionally the proposed approach – MMI – is not only for Persian reviews and in addition can be applied to other domains or other classification problems.

4   Conclusion and Future Works

In this paper we proposed a novel approach for feature selection, MMI, in sentiment classification problem. In addition we applied other feature selection approaches, DF, MI and TFV with the Naive Bayes learning algorithm to the online Persian cellphone

reviews. As the results show, using feature selection in sentiment analysis can improve the performance. The proposed MMI method that uses the positive and negative factors between features and classes improves the performance compared to the other approaches. In our future work we will focus more on sentiment analysis about Persian text.

References

1. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. Mining Text Data. pp. 415–463 (2012)
2. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86. ACL (2002)
3. Moraes, R., Valiati, J.F., Gavião Neto, W.P.: Document-level sentiment classification: an empirical comparison between SVM and ANN. Expert Systems with Applications (2012)
4. Cui, H., Mittal, V., Datar, M.: Comparative experiments on sentiment classification for online product reviews. In: Proceedings of National Conference on Artificial Intelligence, Menlo Park, Cambridge, London, vol. 21(2), p. 1265 (2006)
5. Yussupova, N., Bogdanova, D., Boyko, M.: Applying of sentiment analysis for texts in russian based on machine learning approach. In: Proceedings of Second International Conference on Advances in Information Mining and Management, pp. 8–14 (2012)
6. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (2005)
7. Zhu, J., Wang, H., Zhu, M., Tsou, B.K., Ma, M.: Aspect-based opinion polling from customer reviews. IEEE Transactions on Affective Computing 2(1), 37–49 (2011)
8. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of Conference on World Wide Web, pp. 342–351 (2005)
9. Turney, P.D., Littman, M.L.: Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical Report EGB-1094, National Research Council Canada (2002)
10. Shams, M., Shakery, A., Faili, H.: A non-parametric LDA-based induction method for sentiment analysis. In: Proceedings of 16th IEEE CSI International Symposium on Artificial Intelligence and Signal Processing, pp. 216–221 (2012)
11. Farhoodi, M., Yari, A.: Applying machine learning algorithms for automatic Persian text classification. In: Proceedings of IEEE International Confernce on Advanced Information Management and Service, pp. 318–323 (2010)
12. Taghva, K., Beckley, R., Sadeh, M.: A stemming algorithm for the Farsi language. In: Proceedings of IEEE International Conference on Information Technology: Coding and Computing, ITCC, vol. 1, pp. 158–162 (2005)
13. Mitchell, T.: Machine Learning, 2nd edn. McGraw-Hill (1997)
14. Duric, A., Song, F.: Feature selection for sentiment analysis based on content and syntax models. Decision Support Systems (2012)

Towards the Refinement of the Arabic Soundex

Nedjma Djouhra Ousidhoum and Nacéra Bensaou

University of Sciences and technology Houari Boumedienne
Algiers, Algeria

Abstract. In this paper, we present phonetic encoding functions that play the role of hash functions in the indexation of an Arabic dictionary. They allow us to answer approximate queries that, given a query word, ask for all the words that are phonetically similar to it. They consider the phonetic features of the standard Arabic language and involve some possible phonetic alterations induced by specific habits in the pronunciation of Arabic.

We propose two functions, the first one is called the "Algerian Dialect Refinement" and it takes into account phonetic confusions usually known to the Algerian people while speaking Arabic; and the second one is named the "Speech Therapy Refinement" and it examines some mispronunciations common to children.

1 Introduction

The general goal of approximate string matching is to perform the string matching of a pattern P in a text T where one or both of them have suffered from some kind of corruption [8, 9]

In this paper, we take in interest the problem of approximate string matching that allows phonetic errors. It can be applied in the correction of phonetic spelling errors after a seizure over a speech-to-text system, in the retrieval of similar names or while text searching.

A method to tackle with this issue is to code phonemes using a phonetic encoding algorithm in order to code similarly the words that are pronounced in the same way.

The best-known phonetic encoding algorithm is Soundex [2]. Primarily used to code names based on the way they sound, the Soundex algorithm keeps the first letter of the name, reduces the name to its canonical form [6] and uses three digits to represent the rest of its letters. Many Soundex improvements had been developed for English [18, 7, 11, 12] and it had been expanded for several languages [13, 1] including Arabic [19, 3, 17].

Although spelling correction for Arabic had recently become a very challenging field of research, almost all the solutions in the specialized literature are dedicated to a particular class of Arabic speakers [14]. The same applies to the Arabic Soundex functions which are, moreover, proposed for restricted sets of data such as Arab names [19, 3].

In [10], a Soundex function that takes into account the phonetic features of the Arabic language had been proposed.

In the current work, we define phonetic encoding functions which consist of perfect hash functions that organize the Arabic words into classes of phonetically equivalent ones. Consequently, these functions make possible the indexation of an Arabic dictionary based on phonetics.

We begin by introducing the major phonetic properties of the Arabic language and the process of codifying the Arabic phonemes. Then, the independent functions are exhibited in order to investigate the phonetic alterations which they deal with. Finally, we discuss the statistics of an Arabic dictionary's indexation results and conclude this paper.

2   A Phonetically Indexed Dictionary for Approximate String Matching

2.1   A Phonetic Encoding Function for Arabic

The classical Arabic Soundex [10] is a phonetic encoding algorithm in which phonetic similarity degrees are established based on the rules of the correct reading of Quran [5](which is called Tajweed).

Its classification of phonemes divides the Arabic letters into sets of *phonetic subcategories SC*, in such way that the letters included in the same phonetic subcategory have phonemes that are *very close* phonetically; and the sets of phonetic subcategories that are pronounced from the same area (the tongue, the lips or the throat) are included in the same *phonetic category C*.

For example : Besides using the tongue to pronounce س (s) and ز (z), both

س and ز are produced with air escaping over the sides of the teeth. Therefore,

س and ز have the same phonetic category and the same phonetic subcategory: $C(س) = C(ز)$ and $SC(س) = SC(ز)$.

The Codification of the Classical Arabic Soundex Algorithm. Regarding the phonetic similarity degrees exposed before, the classical Arabic Soundex S attributes to every Arabic letter c a couple of codes, such as $S(c) = (u, v)$ with :

- $u = C(c)$ which is represented by an integer's binary code of two bits,
- $v = SC(c)$ which is represented by an integer's binary code of four bits.

To codify an Arabic word w, first, w must be reduced to its canonical form by a function that deletes identical adjacent letters, blanks and spaces, long vowels and other specific letters from w. Then, S juxtaposes the binary values of the category and the subcategory of each remaining letter of w. Finally, the integer

value of the resulting binary code is computed and the phonetic code generated x can be used as a hash key for indexation.

The indexation of a dictionary D using S generates a hash table where the words that have the same phonetic code k are included in the same set.

For example, given the words $w=$كتف (katif = shoulder) , $w'=$كتب (katab = to write), $w''=$كتم (katam = to hide).

The canonical forms of w, w' and w'' are themselves. Their phonetic codes are then calculated, such as :

$S(w) = C(ك).SC(ك).C(ت).SC(ت).C(ب).SC(ب)$
$= 10.0000.00.0110.00.0001 = 131457 = S(w') = S(w'') = k.$

$Set_k = \{w|S(w) = k\}$, therefore $w, w', w'' \in Set_k$ and w, w', w'' are phonetically equivalent. Set_k fills the cell k of the dictionary D shown below.
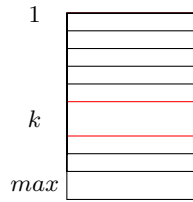

Fig. 1. The phonetically indexed dictionary D

3   Refinement Elements

A refinement element rf is a phonetic encoding function that divides the phonetic categories of Arabic phonemes into new sets of phonetic subcategories regarding a specific pronunciation.

3.1   The Algerian Dialect Function

The "Algerian Dialect Refinement" rf_{alg} is a refinement element that codifies the Arabic letters regarding the confusions between phonemes common to the Algerian Arabic speakers, such as : $rf_{alg}(c) = (N_c, n_c)$ with :

- $N_c = C(c)$,
- n_c is the new phonetic subcategory of c *(Table 1)*.
  Given w a canonical form of a word : $rf_{alg}(w) = N_{c_1}.n_{c_1}.\ldots.N_{c_n}.n_{c_n} = x$.

Table 1. The codification table of the Algerian dialect refinement element

Phonetic Category	Category's code	Subcategory's code	Subcategory's set
Long vowels	Deleted	Deleted	ا و ي (vowel a, o, i)
The letters pronounced using the tongue	0	0	ق، ك (q, k)
		1	ش، ج (sh, j)
		2	ي (غير المدية) (y)
		3	ض، ذ، د ، ظ (dh, th /ô/, d , a strong th /ô/)
		4	س، ز، ص (s, z, a strong 's')
		5	ر، ل، ن (r, l, n)
		6	ت، ث، ط (t, th /theta/, a strong t)
The letters pronounced using the throat	1	0	أ ، ه (e, h)
		1	ع ، ح (these sounds are typical to Arabic)
		2	خ، غ (kh, gh)
The letters pronounced using the lips	2	0	و (غير المدية) (w)
		1	ف، ب، م (f, b, m)

3.2   The Speech Therapy Function

The "Speech Therapy Refinement" rf_{st} is a refinement element that codifies the Arabic letters regarding the phonetic confusions common to children who can't pronounce correctly some Arabic phonemes, such as $rf_{st}(c) = (N_c, n_c)$ with :

- $N_c = C(c)$,
- n_c is the new phonetic subcategory of c *(Table 2)*.

Given w a canonical form of a word: $rf_{st}(w) = N_{c_1}.n_{c_1}.....N_{c_n}.n_{c_n} = x$.

Table 2. The codification table of the speech therapy refinement element

Phonetic Category	Category's code	Subcategory's code	Subcategory's set
Long vowels	Deleted	Deleted	ا و ي
The letters pronounced using the tongue	0	0	ق ، ك (q, k)
		1	ط ، ت ، ث ، س ، ص ، ش
		2	ز، ج، ض، ذ، د، ظ
		3	ي (غير المدية)، ر، ل، ن
The letters pronounced using the throat	1	0	أ ، ه
		1	ع ، ح
		2	خ، غ
The letters pronounced using the lips	2	0	و (غير المدية)
		1	ف، ب، م

4   Evaluation

To assess the phonetic encoding of Arabic using the classical Arabic Soundex S, the Algerian Dialect Refinement rf_{alg} and the Speech Therapy Refinement rf_{st}, we indexed a dictionary of 2017 triliteral Arabic roots (connected to their inflected forms). *Table 3* summarizes the evaluation of this indexation in terms of : distinct codes, words with the same encoding and the maximum value of the encoding.

Table 3. Evaluation table

Algorithm	Number of distinct codes	Maximum number of words with the same encoding	Maximum code
Classical Arabic Soundex Algorithm	872	15	132257
Algerian Dialect Refinement element	766	18	17745
Speech Therapy Refinement element	489	27	2409

4.1   Involving Phonetic Encoding in Spelling Correction

We used our phonetic encoding algorithm in spelling correction by, first, indexing our dictionary. Then, given a query word q, and since Arabic is a highly inflective language [20], we stem q and detect its morphological scheme, we encode its stem r and search for Arabic roots that have the same code of r or approximate ones. Finally, the spelling corrector returns the set of Arabic words that derive from the found roots and have the same morphological scheme of q or similar schemes.

5   Conclusion

Our research contributes to the extension of the Arabic Soundex phonetic encoding algorithm by focusing on specific phonetic criteria related to different sources of phonetic alterations.

This work would help in creating phonetic dictionaries, in resolving the Arabic spelling correction issue by being associated to a spelling corrector like [4, 15, 21, 16] as a module that corrects phonetic spelling mistakes or in detecting sources of confusions between phonemes. It can also be lengthened by supporting new particular "refinement elements".

References

[1] Maniez, D.: Cours sur les Soundex,
    http://www-info.univ-lemans.fr/~carlier/recherche/soundex.html
[2] National Archives: The Soundex Indexing System,
    http://www.archives.gov/research/census/soundex.html

[3] Aqeel, S.U., et al.: On the Development of Name Search Techniques for Arabic. J. Am. Soc. Inf. Sci. Technol. 57(6), 728–739 (2006)

[4] Ben Hamadou, A.: Vérification et correction automatiques par analyse affixale des textes écrits en langage naturel: le cas de l'arabe non voyellé. PhD thesis, University of Sciences, Technology and Medicine of Tunis (2003)

[5] Al Husseiny, A.: Dirassat Qur'aniya-2- Ahkam At-Tajweed Bee Riwayet Arsh An Nafia An Tariq Al'azraq. Maktabat Arradwan (2005)

[6] Hall, P.A.V., Dowling, G.R.: Approximate String Matching. Computing Surveys 12(4) (1980)

[7] Lait, A., Randell, B.: An Assessment of Name Matching Algorithms. Technical Report, University of Newcastle upon Tyne (1993)

[8] Navarro, G.: A Guided Tour to Approximate String Matching. ACM Comput. Surv. 33(1), 31–88 (2001), doi:10.1145/375360.375365

[9] Navarro, G., Baeza-Yates, R.: Very Fast and Simple Approximate String Matching. Information Processing Letters (1999)

[10] Ousidhoum, N.D., Bensalah, A., Bensaou, N.: A New Classical Arabic Soundex algorithm. In: Proceedings of the Second Conference on Advances in Communication and Information Technologies (2012), http://doi.searchdl.org/03.CSS.2012.3.28

[11] Philips, L.: Hanging on the Metaphone. Computer Language 7(12) (December 1990)

[12] Philips, L.: The Double Metaphone Search Algorithm. Dr Dobb's (2003)

[13] Precision Indexing Staff: The Daitch-Mokoto Soundex Reference Guide. Heritage Quest (1994)

[14] Rytting, C.A., et al.: Error Correction for Arabic Dictionary Lookup. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation, LREC 2010 (2010)

[15] Shaalan, K., Allam, A., Gomah, A.: Towards Automatic Spell Checking for Arabic. In: Proceedings of the Fourth Conference on Language Engineering, Egyptian Society of Language Engineering, ELSE (2003)

[16] Shaalan, K., et al.: Arabic Word Generation and Modelling for Spell Checking. In: Proceedings of the Eight International Conference on Language Resources and Evaluation, LREC 2012 (2012)

[17] Shaalan, K., Aref, R., Fahmy, A.: An Approach for Analyzing and Correcting Spelling Errors for Non-native Arabic learners. In: Proceedings of the 7th International Conference on Informatics and Systems, INFOS 2010. Cairo University (2010)

[18] Taft, R.L.: Name Searching Techniques. Technical Report, New York State Identification and Intelligence System, Albany, N.Y. (1970)

[19] Yahia, M.E., Saeed, M.E., Salih, A.M.: An Intelligent Algorithm For Arabic Soundex Function Using Intuitionistic Fuzzy Logic. In: International IEEE Conference on Intelligent Systems, IS (2006)

[20] Watson, J.C.E.: The Phonology and Morphology of Arabic. OUP Oxford (2007)

[21] Ben Othmane Zribi, C., Ben Ahmed, M.: Efficient Automatic Correction of Misspelled Arabic Words Based on Contextual Information. In: Palade, V., Howlett, R.J., Jain, L. (eds.) KES 2003. LNCS, vol. 2773, pp. 770–777. Springer, Heidelberg (2003)

An RDF-Based Semantic Index

F. Amato[1], F. Gargiulo[2], A. Mazzeo[1], V. Moscato[1], and A. Picariello[1]

[1] University of Naples "Federico II", Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, via Claudio 21, 80125, Naples, Italy
[2] Centro Italiano Ricerche Aereospaziali "CIRA" Via Maiorise, 81043, Capua (CE), Italy
{flora.amato,mazzeo,vmoscato,picus}@unina.it,
f.gargiulo@cira.it

Abstract. Managing efficiently and effectively very large amount of digital documents requires the definition of novel indexes able to capture and express documents' semantics. In this work, we propose a novel semantic indexing technique particularly suitable for knowledge management applications. Algorithms and data structures are presented and preliminary experiments are reported, showing the efficiency and effectiveness of the proposed index for semantic queries.

1 Introduction

In this work, we propose a novel *semantic indexing* technique particularly suitable for knowledge management applications. Nowadays, in fact, one of the most challenging aspects in Information Retrieval (IR) area lies in the ability of Information Systems to manage efficiently and effectively very large amount of digital documents by extracting and indexing the related most significant concepts that are generally used to capture and express documents' semantics.

In the literature, the most widely approaches used by IR Systems to allow an efficient semantic-based retrieval on textual documents are: *Conceptual Indexing*, *Query Expansion* and *Semantic Indexing* [1]. All these approaches opportunely combine knowledge representation and natural language processing techniques to accomplish their task. The systems that use the conceptual indexing approach usually ground on catalogs of texts belonging to specific domains and exploit ad-hoc ontologies and taxonomies to associate a conceptual description to documents. In particular, document indexing techniques based on ontology-based concepts' matching are approaches typically used in specialist domains as juridical [2] and medical [3] ones. On the other hand, interesting approaches based on taxonomic relationships are adopted as in [4]: the taxonomic structure is used to organize links between semantically related concepts, and to make connections between terms of a request and related concepts in the index. Differently from the previous ones, systems that use query expansion technique do not need to extract any information from the documents and at the same time to change their structure, but they act on the query provided by the user. The basic idea is to semantically enrich, during the retrieval process, the user query with words that have semantic relationships (e.g. synonyms) with the terms by which the original query is expressed. This approach requires the use of lexical databases and thesauri (e.g. WordNet) and semantic

disambiguation techniques for the query keywords in order to have results that are more accurate [5].

In particular, approaches based on query expansion can be used to broaden the set of retrieved documents, or to increase the retrieval precision using the expansion procedure for adding new terms for refining results. Eventually, systems using semantic indexing techniques exploit the meaning of documents' keywords to perform indexing operations. Thus, semantic indexes include word meanings rather than the terms contained in documents. A proper selection of the most representative words of the documents and their correct disambiguation is indispensable to ensure the effectiveness of this approach. In [6], several interesting experiments of how to use word sense disambiguation into IR systems are reported. A large study about the applicability of semantics to IR is in the opposite discussed in [7] , in which the problem of lexical ambiguity is bypassed associating a clear indication of word meanings to each relevant terms, explicating polysemy and homonymy relationships. Furthermore, a semantic index is built on the base of a disambiguated collection of terms in the SMART IR System designed by [8]. The use of these approaches is obviously limited by the need of having available specific thesauri for establishing the correct relationships among concepts.

In this paper, we describe a semantic indexing technique based on RDF (Resource Description Framework) representation of the main concepts of a document. With the development of the Semantic Web, in fact, a large amount of RDF native documents are published on the Web and, for what concerns digital documents, several techniques could be used to transform a text document into a RDF model, i.e. a subject, verb, object triple [9]. Thus, in our approach, we propose to capture the semantic nature of a given document, commonly expressed in Natural Language, by retrieving a number of RDF triples and to semantically index the documents on the base of meaning of the triples' elements (i.e. subject, verb, object). The proposed index can be hopefully exploited by actual web search engines to improve the retrieval effectiveness with respect to the adopted query keywords or for automatic topic detection tasks.

The paper is organized as in the following. In the next section we illustrate our proposal for RDF based semantic index discussing indexing algorithms and providing some implementation details. Section 3 contains experimentation aiming at validating the effectiveness and efficiency of our proposal. Finally, some conclusions are outlined in Section 4.

2   A Semantic Index Based on RDF Triples

The proposed semantic index relies on a two levels indexing data structure for RDF triples that is built in a bottom-up way. This section describes the algorithms that allow to generate such a data structure and to perform semantic queries.

In our formulation, an RDF triple is a statement which relates a *subject* to an *object* by means of a *predicate*, as in *<Pope, give, Resignation>*. Of course, in some triplets the object could be empty but for the sake of simplicity and without loss of generality, in our model the object is assumed always to be a not-empty entity.

Furthermore, we assume that the considered lexical database contains all subjects, predicates and objects related to all the triples. As consequence, it is always possible to measure the semantic similarity/distance between two elements of a given triple [10].

In particular, the lexical database used in this work is *Wordnet*, while the semantic similarity measure adopted is the *Leachock and Chodorow* metric [11], but the described approach is parametric with respect to the established similarity measure. In the case of multiple senses, words are opportunely disambiguated choosing the most fitting sense for the considered domain using a context-aware and taxonomy-based approach [12], if necessary. Moreover, it is assumed that evaluation of the similarity measure require a constant time. The distance between two RDF triples is defined as a linear combination of the distances between the subjects, the predicates and the objects of the two triples and it also requires a constant time. Finally, it is assumed that the maximum number of clusters m is much smaller than the number of triples N. On the base of such hypothesis, the main steps of the index building algorithm are the following:

1. Create h clusters of triples, with $h \leq m$
2. Create a new cluster considering the centroids of the previous clusters and find the centroid for the new cluster.
3. For each cluster:
   (a) Map each triple of the cluster in a point of R^3.
   (b) Build a k-d TREE ($k = 3$) with the points obtained in previous step.
4. Repeat steps *3(a)* and *3(b)* for the cluster of centroids.

In step one, it is used a *single-pass iterative* clustering method that randomly choose the first triple and create the first cluster, labeling such a triple as the centroid of the first cluster. Successively, the clustering algorithm performs a loop until all input triples are processed in a random order [1]. In particular, for each triple, the clustering algorithm tries to find the most suitable cluster and then adds the triple to the cluster: the most suitable cluster is the cluster for which the centroid is closest to the current triple. If such distance is less than the threshold t then the algorithm adds the current triple to the most suitable cluster. Otherwise, if the current number of clusters is less then maximum number of cluster m then a new cluster is created and the algorithm marks the current triple as the centroid of the new cluster itself. In the opposite, if a new cluster cannot be created, because the current number of cluster is equal to m, the current triple is added to the most suitable cluster, even if the distance between the triple and the centroid is greater than the threshold. The complexity of the clustering algorithm is $O(mN)$ because for each triple, the algorithm evaluates at most m distances. The second step is devoted to build a new cluster, considering the centroids of the previous clusters, and to find the correct centroid for the new cluster as the triple having the minimum average distance from the other ones. In particular, this step requires an $O(m)$ time for the creation of the cluster and an $O(m^2)$ time to find the centroid of the new cluster. The third step performs a loop over all the clusters obtained in the first step and is composed by two sub-steps. In sub-step 3(a) the algorithm maps each triple in a point of R^3. The x-coordinate of the point is the distance between the subject of the triple and the subject

---

[1] For the semantic clustering aims, we can select any supervised clustering algorithm able to partition the space of documents into several clusters on the base of the semantic similarity among triples.

of the centroid of the cluster. In the same way, the algorithm calculates the y-coordinate and z-coordinate using the predicates and objects elements.

In sub-step 3(b) the algorithm builds a *3-d tree* [13] with the obtained mapped points. The overall complexity of the third step is $O(N \log N)$, that is time required to build the 3-d trees. The fourth step repeats the sub-step 3(a) with the cluster of centroids obtained in the second step and the sub-step 3(b) with the related mapped points. The complexity of the step 4 is $O(m \log m)$ because the cluster of centroids contains at most m triples. Hence, the presented algorithm builds the data structure behind the semantic index in an $O(N \log N)$ time, where N is the number of RFD triples. The $O(m^2)$ time is dominated by $O(N \log N)$ because m is much smaller than N.

Figure 1 shows an example of a part of our indexing structure in the case of web pages reporting some news on latest events happening in Italy (e.g. a search performed using different combination of query triples will produce a result set containing the documents or parts of them which main semantics effectively corresponds to the query one). At the first level, we adopt a 3-d tree to index the triples related to the centroids of the clusters, while at second one, a 3-d tree is used for each cluster and the triples can contain a reference to the original document from which they came from.
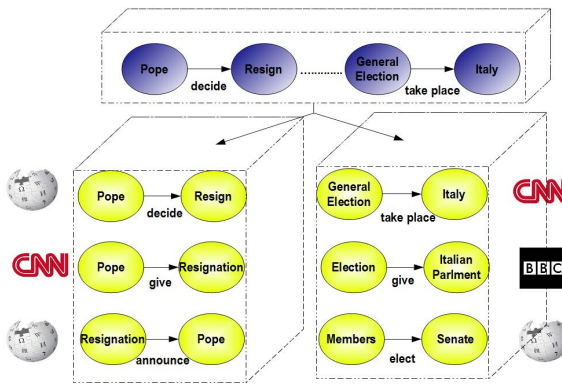

Fig. 1. Example of Semantic Index

Given an RFD query triple q, the execution of a semantic query is accomplished through the following steps:

1. Map q in R^3
2. Find the closest centroid c to q in the 3-d tree of the cluster of centroids
3. Find the closest point p to q in the 3-d tree of the cluster C (C is the cluster realted to the centroid c)
4. Return the RFD triple associated to p

In a similar way, a *range query* and *k-nearest neighbors* query can be performed on our indexing structure. Hence, this kind of search can be done efficiently by using the well-known k-d tree properties.

3   Preliminary Experimental Results

In this section, we describes the adopted experimental protocol, used to evaluate the efficiency and effectiveness of our indexing structure, and discussing the obtained preliminary experimental results.
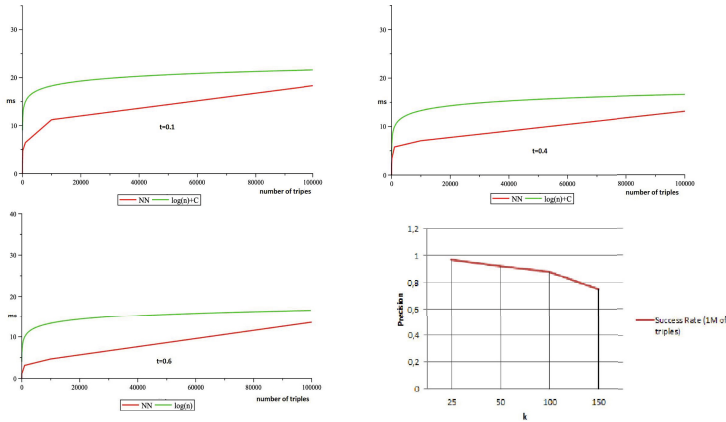

Fig. 2. Average Search Times and Average Success Rate using the Semantic Index

Regarding the triples collection, we selected a subset of the **Billion Triple Challenge 2012 Dataset**[2], in which data are encoded in the *RDF NQuads* format, and used the *context* information to perform a correct semantic disambiguation of triples elements[3]. In particular, as evaluation criteria in the retrieval process using our semantic index, we measured from one hand the *average search time* as a function of indexed triples, and from the other one the *success rate*, in other terms the number of *relevant*[4] returned triples with respect to several performed k-nearest neighbors queries on the data collection. The obtained results were studied using different values for the clustering threshold t (0.1,0.4,0.6) and for k (the result set size).

The Figure 2 shows the average search times for the different values of t. The search time function exhibits in each situation a logarithmic trend and the asymptotic complexity is $O(log(n) + c)$, n being the number of triples, as we theoretically expected. For what the effectiveness concerns, we have computed a sort of average precision of our index in terms of relevant results with respect to a set of query examples (belonging to different semantic domains).

---

[2] `http://km.aifb.kit.edu/projects/btc-2012/`

[3] The Billion Triple Challenge 2012 dataset consists of over a billion triples collected from a variety of web sources in the shape $< subject >< predicate >< object >< context >$ (e.g. $< pope >< decide >< resign >< religion >$). The dataset is usually used to demonstrate the scalability of applications as well as the capability to deal with the specifics of data that has been crawled from the public web.

[4] A result triple is considered relevant if it has a similar semantics to the query triple.

In particular, a returned triple is considered *relevant* if it belongs to the same semantic domain of the query triple. The same Figure shows the obtained results for the average success rate varying the result set size in the case of 50 queries performed on the entire dataset (about 1000000 triples) and using an index generated with $t = 0.6$.

4   Conclusion and Future Works

In this paper, we have designed a semantic index based on RDF triples, in order to catch and manage the semantics of documents. We have described the data structures, the algorithms for building the index and its use for semantic queries. Several preliminary experiments have been carried out using the standard Billion Triple Challenge 2012 Data Set, showing good performances both for efficiency and for effectiveness. We are planning to extend our paper in several directions: i) the use of domain based linguistic ontologies, instead or in addition to the used WordNet; ii) the use of different similarity distance measures; iii) to compare our algorithms with the several ones produced in the recent literature.

References

1. Mihalcea, R., Moldovan, D.: Semantic indexing using wordnet senses. In: Proceedings of the ACL Workshop on IR and NLP, Hong Kong (2000)
2. Stein, J.: Alternative methods of indexing legal material: Development of a conceptual index. In: Conference Law Via the Internet 1997, Sydney, Australia (1997)
3. Amato, F., Fasolino, A., Mazzeo, A., Moscato, V., Picariello, A., Romano, S., Tramontana, P.: Ensuring semantic interoperability for e-health applications. In: 2011 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), pp. 315–320. IEEE (2011)
4. Woods, W.: Conceptual indexing: A better way to organize knowledge (1997)
5. Mihalcea, R., Moldovan, D.: An iterative approach to word sense disambiguation. In: Proceedings of FLAIRS, pp. 219–223 (2000)
6. Mihalcea, R., Moldovan, D.: Semantic indexing using wordnet senses. In: Proceedings of the ACL 2000 Workshop on Recent Advances in Natural Language Processing and Information Retrieval, Association for Computational Linguistics, pp. 35–45 (2000)
7. Stokoe, C.: Differentiating homonymy and polysemy in information retrieval. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 403–410 (2005)
8. Gonzalo, J., Verdejo, F., Chugur, I., Cigarran, J.: Indexing with wordnet synsets can improve text retrieval. arXiv preprint cmp-lg/9808002 (1998)
9. D'Acierno, A., Moscato, V., Persia, F., Picariello, A., Penta, A.: iwin: A summarizer system based on a semantic analysis of web documents. In: 2012 IEEE Sixth International Conference on Semantic Computing (ICSC), pp. 162–169 (2012)
10. Hirst, G., Mohammad, S.: Semantic distance measures with distributional profiles of coarse-grained concepts. In: Modeling, Learning and Processing of Text Technological Data S. (2012)
11. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. WordNet: An Electronic Lexical Database 49, 265–283 (1998)
12. Mandreoli, F., Martoglia, R.: Knowledge-based sense disambiguation (almost) for all structures. Information Systems 36, 406–430 (2011)
13. Samet, H.: The design and analysis of spatial data structures, vol. 85. Addison-Wesley, Reading (1990)

Experiments in Producing Playful "Explanations" for Given Names (Anthroponyms) in Hebrew and English

Yaakov HaCohen-Kerner[1], Daniel Nisim Cohen[1], and Ephraim Nissan[2]

[1] Dept. of Computer Science, Jerusalem College of Technology, 91160 Jerusalem, Israel
kerner@jct.ac.il, sdanielco@gmail.com
[2] Dept. of Computing, Goldsmiths' College, Univ. of London, 25–27 St. James, New Cross,
London SE14 6NW, England, United Kingdom
ephraim.nissan@hotmail.co.uk

Abstract. In this project, we investigate the generation of wordplay that can serve as playful "explanations" for given names. We present a working system (part of work in progress), which segments and/or manipulates input names. The system does so by decomposing them into sequences (or phrases) composed of at least two words and/or transforming them into other words. Experiments reveal that the output stimulates human users into completing explanations creatively, even without sophisticated derivational grammar. This research applies to two languages: Hebrew and English. The applied transformations are: addition of a letter, deletion of a letter and replacement of a similar letter. Experiments performed in these languages show that in Hebrew the input and output are perceived to be reasonably associated; whereas, the English output, if perceived to be acceptable rather than absurd, is accepted as a humorous pun.

Keywords: Computational Humour, English, Folk-etymology, Hebrew, Mock-aetiology (playful explanations), Onomastics, Puns, Wordplay.

1   Introduction

"Wordplay is a literary technique and a form of wit in which the words that are used become the main subject of the work, primarily for the purpose of intended effect or amusement. Puns, phonetic mix-ups such as spoonerisms, obscure words and meanings, clever rhetorical excursions, oddly formed sentences, and telling character names are common examples of word play".[1] Pragmaticians and literary scholars have researched puns [1-4]. Pun-generating software exists [5-6]. Software tools for entertainment are occasionally intended to stimulate human cognition as well as to make the user experience a gratifying playful experience. This is true for Serendipity Generators, apps which suggest to a user which way to turn when out for a stroll [7].

We are primarily interested in an application to Hebrew wordplay, and this requires a fresh look. Within computational humour, automatically devising puns or

---

[1] http://en.wikipedia.org/wiki/Word_play

punning riddles as figured prominently [5-6]. Our main contribution in this study has been to present how Hebrew wordplay differs from wordplay in other languages (English).

Rabbinic homiletics revels in wordplay that is only sometimes humorous. A poetic convention enables gratification that is not necessarily humorous. Cultural exposure to this tradition apparently conditions human appreciation of the Hebrew outputs of our tool, so that humour is not necessarily a requirement for gratification from the playfulness deriving from those outputs. As in other computational humour tools, we do not have a model of humour in our software, which assists users in experiencing gratification from onomastic wordplay. The input of the software is a personal given name (a forename or a first name) Our working system either segments an input name, and/or introduces minimal modifications into it, so that the list of one or more components are extant words (Hebrew if the input is Hebrew, English if the input is English), whose juxtaposition is left to the user to make sense of. As it turns out, such output stimulates creativity in the subjects (ourselves) faced with the resulting word-lists: they often easily "make sense". We segment and/or use transformations such as: addition of a letter, deletion of a letter, and a replacement of a similar letter.

These are graphemic puns [8]. These, using letter replacements, have been previously applied in DARSHAN [9]. DARSHAN generates ranked sets of either one-sentence or one-paragraph homilies using various functions, e.g., pun generations, numerological interpretations, and word or letter replacements. Our next step would be to start to implement the punning module of GALLURA, at present a theoretical model that generates playful explanations for input proper names (such as place-names) by combining phono-semantic matching (PSM) with story-generation skills [10-12]

Hebrew is a Semitic language. It is written from right to left. Inflection in Semitic, like in Romance, is quite rich, but Semitic morphology is nonconcatenative (with the consonants of the root being "plugged" into "free spaces" of a derivational or inflectional pattern). [13] is a survey of Hebrew computational linguistics. It is important to note that the very nature of the Hebrew script is somewhat conducive to success: it is a consonantal script, with some letters inserted in a mute function (*matres lectionis*) suggesting vowels: w is [v] or [o] or [u]; y is consonantal [y] or vocalic [i] or long [e].

Section 2 presents the workings of the model. Section 3 presents the results of experiments and analyzes them. Section 4 provides several illustrative examples and their analysis. Section 5 concludes the paper, and proposes a potential future research.

2   The Model

Given a name, our system tries to propose one word or a sequence of words as a possible playful, punning "explication" in the same language as the input name. The output should be rather similar to the input word from the spelling viewpoint. These are "graphemic puns" indeed. To generate similar word(s) from the spelling viewpoint, we divide the given word into relevant sequential sequences of sub-words,

which compose the input word and/or apply one or two of the following three transformations: deletion of a letter, insertion of a letter, replacement of a similar letter.

In order to avoid straying from the given names, we have performed up to 2 transformations on each given name (i.e. according to Levenshtein measure, the maximal allowed distance is 2).

A similar letter in Hebrew is replaced using groups of Hebrew letters that either sound similar or are allographs of the same grapheme. Examples of such groups are: א ע - ה -, ב - ו , ת - ט, ך - ק - כ, ם - מ , ן - נ, ף - פ, ץ - צ - ז , and ס – ש.

Examples of groups of English letters that sound similar: a - e (e.g., see, sea; man, men), a - w (e.g., *groan, grown*), b - p (e.g.: *bin, pin*), d - t (e.g., *bead, beat*), e - w (e.g., *shoe, show*), f - p (e.g., *full, pull*), m - n (e.g., *might, night*), o - w (e.g., *too, two*), s - z (e.g., *analyse, analyze*), and u - w: (e.g., *suite, sweet*). Except the use of these groups of similar letters, we have no phonetic model in our software.

3   Experiments

Experiments have been performed in two languages: Hebrew and English. For each language, we used two main datasets: a list of given names and a lexicon (i.e. the language's vocabulary). Table 1 shows general details about these four datasets.

Table 1. General details about the datasets used

Name of Data Set	# of different words	Source
Given names in Hebrew	1,758	`http://www.babyworld.co.il/`
Given names in English	1,992	`http://www.thinkbabynames.com/popular/0/us/1`
Words in Hebrew	257,483	Bar-Ilan University's Corpus of Modern Hebrew
Words in English	496,688	`http://www.wordfrequency.info/intro.asp`

We have chosen 50 names in Hebrew and 50 names in English for which our system seems to produce somewhat "surprising" associations. Each one of the results has been evaluated manually by two people who are speakers of both Hebrew and English.

Three evaluation criteria were required for each output: grammatical correctness, "creative" level, and sound similarity to the original (input) word. For each criterion, the reviewer was required to give an evaluation from 5 (the highest) to 1 (the lowest). The value of the grammatical correctness measure represents the degree of the grammatical correctness of the produced explanation. For instance, if the "explanation" produced is a phrase containing at least two words, the evaluation is given according to the grammatical connections between the words. The value of the creative level measure represents the degree of creativity of the produced explanation in regards to surprise and novelty. The value of the sound similarity measure represents the degree of the sound similarity between the produced explanation and

the original word when they are pronounced. Tables 2 and 3 present the values of these three evaluation criteria for the produced explanations in Hebrew and English, respectively.

Table 2. General statistics of produced explanations for 50 names in Hebrew

Number of transformations	Number of words	Grammatical correctness			Creative level			Sound similarity		
		avg	med	std	avg	med	std	avg	med	std
0 (only decomposition)	8	4.56	5	0.56	2.75	3	1.34	3.69	4	1.22
1	18	4.77	5	0.16	2.74	3	0.18	3.19	3	0.15
2	24	4.38	5	0.44	3.62	4	0.25	2.39	2	0.85

Table 3. General statistics of produced explanations for 50 names in English

Number of transformations	Number of words	Grammatical correctness			Creative level			Sound similarity		
		avg	med	std	avg	med	std	avg	med	std
0 (only decomposition)	10	3.25	3	1.25	2.6	3	0.88	3.85	4	0.77
1	22	3.3	3.5	0.28	2.83	3	0.14	2.55	3	0.19
2	18	3.12	3.5	0.52	2.94	2	0.39	2.84	3	0.26

The comparison between the grammatical correctness evaluations of the Hebrew outputs (around 4.5-4.6) and the English outputs (around 3.2-3.3) clearly shows that the Hebrew outputs are perceived to be much more correct grammatically. The creative level evaluations of the Hebrew outputs and the English outputs yielded similar results. In both languages, the more transformations carried out, the higher the "creative" level. This finding is reasonable since activating more transformations increases the differences between the original input word and the output, raising the level of creativity.

The values of the sound similarity measure present almost an opposite picture. That is to say, in both languages, in general, the more transformations are carried out the lower the sound similarity between the input and the output. The low values of the standard deviation in Tables 2 and 3 indicate that the results of the average and median in both languages are stable and reliable.

4   Illustrative Examples

Tables 4 and 5 present six detailed examples for Hebrew and English, respectively.

Due to space limitations, we shall explain in detail only one relatively complex example. The input of the fifth Hebrew example is the Hebrew name אביתר (Evyatar, Ebiatar). Firstly, the system segments the input into a sequence of two acceptable words in Hebrew and then activates two letter additions. The output is the following sequence: יותר (2) אהב (1), which means "he loved more".

Table 4. Examples for results in Hebrew

#	Input	Trans. (1)	Trans. (2)	Trans. (3)	Output
1	אבינתן	Segment: נתן 2 אבי 1			נתן 2 אבי 1
2	חמוטל	Segment: טל 3 ו 2 חמ 1	Replace: טל 3 ו 2 חם 1		טל 3 ו 2 חם 1
3	חבקוק	Delete: חקוק			חקוק 1
4	אביטוב	Add: אהביטוב	Delete: אהבטוב	Segment: טוב 2 אהב 1	טוב 2 אהב 1
5	אביתר	Segment: אב 2 יתר 1	Add: יתר 2 אהב 1	Add: יותר 2 אהב 1	יותר 2 אהב 1
6	סחלב	Segment: לב 2 סח 1	Replace: לב 2 שח 1	Add: ללב 2 שח 1	ללב 2 שח 1

Table 5. Examples for results in English

#	Input	Trans. (1)	Trans. (2)	Output
1	Beatrice	Segment: 1 Beat 2 rice		1 Beat 2 rice
2	Cameron	Del: Cameon	Segment: 1 Came on	1 Came 2 on
3	Payton	Del: Payon	Segment: 1 Pay 2 on	1 Pay 2 on
4	Kingston	Add: Kingstone	Segment: 1 King 2 stone	1 King 2 stone
5	Cannon	Replace: Cannot	Segment:1 Can 2 not	1 Can 2 not
6	Tobias	Replace: Dobias	Segment:1 Do 2 bias	1 Do 2 bias

Table 5 presents examples in English that were produced automatically by our system. It is evident, from this table, that the segmentations and modifications were performed on the graphemic string, without involving phonology, but the impact of this is quite different than with the experiments conducted in Hebrew. The outputs in English are not as successful as those obtained in Hebrew (this is itself an interesting, albeit foreseeable result), and this is because whereas the Hebrew script is consonantal, and readers are left to fill in the vowels, English spelling is almost more puzzling, because of the irregularity of the correspondence between the written word and pronunciation. Mastering this is crucial to learning English as spoken and as written.

Tables 6 and 7 present the values of the 3 evaluation criteria, provided by the two reviewers (native speakers of Hebrew, not English), for the "(homiletic) explanations" produced for the input words mentioned in tables 4 and 5, respectively. The average values in the last rows of tables 6 and 7 show that: (a) the grammatical correctness evaluations of the Hebrew outputs (5) are appreciably higher than those of the English outputs (4.17-4.5), (b) the creative level evaluations of the Hebrew outputs (2.5-3.17) are lower than those of the English outputs (3-3.67); a possible explanation to this surprising finding is that "creative" "explanations" are in many cases based on exceptions from grammatical correctness and sound similarity, and (c) the sound similarity evaluations of the Hebrew outputs (3.67-4) are higher than those of the English outputs (3.33-3.5).

Table 6. Evaluations conducted by the two reviewers for 6 examples in Hebrew

#	Input word	Output	Evaluation of the first reviewer			Evaluation of Second reviewer		
			grammatical correctness	creative level	sound similarity	grammatical correctness	creative level	sound similarity
1	אבינתן	נתן 2 אבי 1	5	1	5	5	1	5
2	חמוטל	טל 3 ו 2 חם 1	5	3	4	5	5	3
3	חבקוק	חקוק 1	5	1	4	5	2	4
4	אביטוב	טוב 2 אהב 1	5	3	3	5	3	3
5	אביתר	יותר 2 אהב 1	5	3	3	5	3	2
6	סחלב	ללב 2 שח 1	5	4	5	5	5	5
	averages:		5	2.5	4	5	3.17	3.67

Table 7. Evaluations conducted by the two reviewers for 6 examples in English

#	Input word	Output	Evaluation of the first reviewer			Evaluation of Second reviewer		
			grammatical correctness	creative level	sound similarity	grammatical correctness	creative level	sound similarity
1	Beatrice	1 Beat 2 rice	4	4	3	3	4	4
2	Cameron	1 Came 2 on	4	3	3	4	4	3
3	Payton	1 Pay 2 on	5	3	3	5	4	3
4	Kingston	1 King 2 stone	5	2	5	5	3	5
5	Cannon	1 Can 2 not	5	4	4	5	4	3
6	Tobias	1 Do 2 bias	4	2	3	3	3	2
	averages:		4.5	3	3.5	4.167	3.67	3.33

5   Summary and Future Research

We have presented a system that, when fed a one-word input (a personal given name) segments it and/or modifies it using one or two transformations (addition of a letter, deletion of a letter and replacement of a similar letter) so that the output is a list of words extant in the lexicon of the same language as the input. Our experiments show that in Hebrew reasonable association between the input and output is perceived as higher than in English. As for English, often the segmentation or modification is perceived to be underwhelming, but on occasion containing some element of surprise. Arguably, the nature of both Hebrew writing and Hebrew morphology militate towards such differences in perception. However, cultural factors also contribute to make the associations proposed by the tool for Hebrew, more readily accepted by members of the culture, not necessarily as a joke. English output, if perceived to be acceptable rather than absurd, is accepted instead as a (mildly) humorous pun.

We have contrasted our kind of graphemic puns to ones from the Far East [8]. In addition, we have designed a phono-semantic matching (PSM) module [10] interfacing a future story-generation tool for devising playful explanations for input proper names [11-12]. This phenomenon is known from human cultures in various contexts, e.g. [14].

References

1. Redfern, W.D.: Puns. Basil Blackwell, Oxford (1984); 2nd edn. Penguin, London (2000)
2. Sharvit, S.: Puns in Late Rabbinic Literature. In: Schwarzwald, O.R., Shlesinger, Y. (eds.) Hadassah Kantor Jubilee Volume, pp. 238–250. Bar-Ilan University Press, Ramat-Gan (1995) (in Hebrew)
3. Sharvit, S.: Play on Words in Late Rabbinic Literature. In: Hebrew Language and Jewish Studies, Jerusalem, pp. 245–258 (2001) (in Hebrew)
4. Dynel, M.: How do Puns Bear Relevance? In: Kisielewska-Krysiuk, M., Piskorska, A., Wałaszewska, E. (eds.) Relevance Studies in Poland. Exploring Translation and Communication Problems, vol. 3, pp. 105–124. Warsaw Univ. Press, Warsaw (2010)
5. Hempelmann, C.F.: Paronomasic Puns: Target Recoverability towards Automatic Generation. PhD thesis. Purdue University, Indiana (2003)
6. Waller, A., Black, R., Mara, D.A.O., Pain, H., Ritchie, G., Manurung, R.: Evaluating the STANDUP Pun Generating Software with Children with Cerebral Palsy. ACM Transactions on Accessible Computing (TACCESS) 1(3), article no. 16, at the ACM site (2009)
7. de Lange, C.: Get Out of the Groove. New Scientist 215(2879), 47–49 (2012)
8. HaCohen-Kerner, Y., Cohen, D.N., Nissan, E., Zuckermann, G.: Graphemic Puns, and Software Making Them Up: The Case of Hebrew, vs. Chinese and Japanese. In: Felecan, O. (ed.) Onomastics in the Contemporary Public Space. Cambridge Scholars Publishers, Newcastle (in press)
9. HaCohen-Kerner, Y., Avigezer, T.S., Ivgi, H.: The Computerized Preacher: A Prototype of an Automatic System that Creates a Short Rabbinic Homily. B.D.D. (Bekhol Derakhekha Daehu): Journal of Torah and Scholarship 18, 23–46 (2007) (in Hebrew)
10. Nissan, E., HaCohen-Kerner, Y.: The Design of the Phono-Semantic Matching (PSM) Module of the GALLURA Architecture for Generating Humorous Aetiological Tales. In: Felecan, O. (ed.), Unconventional Anthroponyms. Cambridge Scholars Publishers, Newcastle (in press)
11. Nissan, E., HaCohen-Kerner, Y.: Information Retrieval in the Service of Generating Narrative Explanation: What we Want from GALLURA. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Information Retrieval (KDIR), pp. 487–492 (2011)
12. Nissan, E., HaCohen-Kerner, Y.: Storytelling and Etymythology: A Multi-agent Approach (A Discussion through Two "Scandinavian" Stories). In: HaCohen-Kerner, Y., Nissan E., Stock, O., Strapparava, C., Zuckermann, G. (eds.), Research into Verbal Creativity, Humour and Computational Humour. Topics in Humor Research, Benjamins, Amsterdam (to appear)
13. Wintner, S.: Hebrew Computational Linguistics: Past and Future. Artificial International Review 21(2), 113–138 (2004)
14. Zuckermann, G.: "Etymythological Othering" and the Power of "Lexical Engineering" in Judaism, Islam and Christianity. In: Omoniyi, T., Fishman, J.A. (eds.) Explorations in the Sociology of Language and Religion, Benjamins, Amsterdam, ch.16, pp. 237–258 (2006)

Collaborative Enrichment of Electronic Dictionaries Standardized-LMF

Aida Khemakhem[1], Bilel Gargouri[1], and Abdelmajid Ben Hamadou[2]

[1] MIRACL Laboratory, University of Sfax
FSEGS, B.P. 1088, 3018 Sfax, Tunisia
[2] MIRACL Laboratory, University of Sfax
ISIMS, B.P. 242, 3021 Sakiet-Ezzit Sfax, Tunisia
{aida.khemakhem,bilel.gargouri}@fsegs.rnu.tn,
abdelmajid.benhamadou@isimsf.rnu.tn

Abstract. The collaborative enrichment is a new tendency in constructing resources, notably electronic dictionaries. This approach is considered very efficient for resources weakly structured. In this paper, we deal with applying the collaborative enrichment for electronic dictionaries standardized according to LMF-ISO 24613. The models of such dictionaries are complex and finely structured. The purpose of the paper is, on the one hand, to expose the challenges related to this framework and, in the second hand, to propose practical solutions based on an appropriate approach. This approach ensures the properties of completeness, consistency and non-redundancy of lexical data. In order to illustrate the proposed approach, we describe the experimentation carried out on a standardized Arabic dictionary.

Keywords: Collaborative enrichment, LMF normalized dictionaries, coherence, non-redundancy, completeness.

1   Introduction

Electronic dictionaries contribute enormously to the learning, the dissemination, the maintenance and the evolution of natural languages. However, the construction of such dictionaries is a difficult task given the richness of natural languages. It is very expensive in time and number of people typing dealing with enormous content of lexical resources. Moreover, it is not limited in time because of the continuous need of enrichment.

In order to tackle the problems related to the enrichment of electronic dictionaries, the tendency was the resort to a collaborative approach. Therefore, several works was proposed such as [3] [4] [12] and [14]. The well known application of the collaborative approach for filling and updating large resources is the Wiktionary [13] that currently covers several languages. However, the mentioned works deal with a superficially structure (or model) of resources. Indeed, their syntactic models are very light and don't link synonyms through the concerned senses. Moreover, relation between senses and syntactic knowledge are not covered. Thus, the update of such resources is available for all kinds of users who are not necessarily experts in the lexicography or in the linguistic domains.

Few years ago, the dictionaries construction field has been consolidated by the publication of the LMF-ISO 24613 norm (Lexical Markup Framework) [6]. This norm offers a framework for modeling large lexical resources in a very refined way. LMF has been proved compatible with the majority of vehicular languages.

In this paper, we highlight the challenges related to the collaborative enrichment of large, LMF-standardized dictionaries. In such dictionaries, enormous knowledge are required and can be defined only by expert users (i.e., linguists, lexicographers) such as syntactic behaviors, semantic roles and semantic predicates. In addition, specific links are to be considered among several lexical entries such as morphological derivation links or semantic relations links (i.e., synonymy, antinomy) between senses. Other more complex links are as those of syntactic-semantic dependency. The enrichment difficulty increases when the partners of a link are entered in the dictionary separately. As difficulty, we can mention the case of the synonymy link where the corresponding sense to be linked is not yet introduced. In general, the issues affecting the integrity of the dictionary concern the absence of mandatory knowledge, a wrong links and redundant knowledge.

The solution that we propose is a wiki-based approach that benefits from the fine structure ensured by LMF. This fine structure provides selective access to all knowledge in the normalized dictionary and consequently promotes the control of the enrichment. The proposed approach ensures the properties of completeness, coherence and non-redundancy using a set of appropriate rules. In order to illustrate the proposed approach, we give a report on the experimentation carried on an Arabic normalized dictionary [9].

We start with giving an overview of the main approaches used for the enrichment of electronic dictionaries. Then, we introduce the LMF norm. Thereafter, we describe the main issues related to the collaborative enrichment of the LMF-standardized dictionaries. After that, we expose the proposed approach. Afterward, we detail the application of the proposed approach on an Arabic normalized dictionary.

2   Enrichment Approaches of Electronic Dictionaries

In this section we try to enumerate the main approaches that were proposed to accomplish the enrichment of electronic dictionaries. The first one is based on the massive typing of the content of one or several paper dictionaries. This was the case for example of the dictionary TLFi [5], using the sixteen paper volumes of the old dictionary TLF ("Trésor de la Langue Française") in the Atilf laboratory in French. This approach is considered very costly in time and number of people typing, although it provides reference content.

In order to reduce the cost of the typing, some works have recourse to the digitization of old dictionaries. This was the case for example of several dictionaries of Arabic as implemented in sakhr environment[1]. This approach is inexpensive but provides unstructured dictionaries and therefore research services are very rudimentary.

---

[1] `lexicons.sakhr.com`

Moreover, some researchers have suggested using (semi) automatic conversion content of numeric version in a format that can be processed by a machine (eg, txt, html). Among these works we can mention the DiLAF project for the Africans-French languages [11], the Euskal Hiztegia (EH) dictionary of Basque [1] and the work of converting the dictionary Al-Ghany in an LMF standardized template for Arabic [10]. The main weakness of this approach is the high rate of recognition failure for several Lexical Entry (LE) knowledge.

Lately, the trend was a wiki or collaborative solution. Among the well known examples, we mention the Wiktionary [13]. Any user can contribute to the enrichment phase through a previously authentication. Other users having elevated permissions can remove some user's suggestions if they estimate their non-validity. The wiki solution is quite efficient to obtain a huge content of knowledge in continuous update but it has some drawbacks. Indeed, the handled knowledge are linguistically superficial what explains first the weakness of the structure and second the recourse to unskilled users. Moreover, the control of propositions is mainly based on human expertise.

We note that the collaborative constructed resources can also be combined with others structured resources to construct a large scale dictionaries like in the Uby project [8].

3   LMF Standard vs Collaborative Enrichment

3.1   General Presentation of LMF

The Lexical Markup Framework (LMF) [2] is published in 2008 as an ISO 24613 standard for the modeling of lexical resources usable for the editorial needs as well as for the Natural Language Processing (NLP) ones. It proposes a meta-model composed by a core and optional extensions (i.e., morphological, syntactic, semantic).

Apart from a variety of knowledge that one can represent, several kinds of links are to be considered. For example, to ensure links between a derived form and its stem, we use the class *RelatedForm* with the attribute *type* to indicate the kind of the morphological link. In the Figure 1, we show the links *derivedForm* and *stem* between the two lexical entries *write* and *writer*.

As syntactic-semantic links, we quote the possible frames of a LE and for each frame we denote the related senses through the Syntactic Behavior as shown in Figure 2. Regarding semantic links, we mention the use of the *SenseRelation* class that allows connecting senses belonging to different LEs.
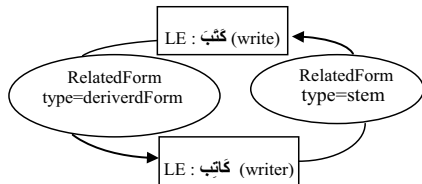

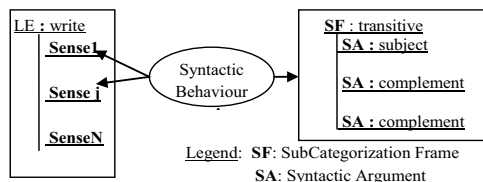**Fig. 1.** Example of a morphological links   **Fig. 2.** Example of a syntactic-semantic links

3.2   Impact of LMF on the Collaborative Enrichment

LMF allows building dictionaries with a large linguistic coverage. Thus, the knowledge to be introduced for a single lexical entry are various and leads relationships with other entries or directly with knowledge of other entries. Such information and links are not simple and require linguistic expertise. Consequently, the enrichment phase, notably with a collaborative approach might be a difficult task.

From another point of view, LMF has its advantages that favor the use of a collaborative enrichment and thus reduce the complexity of this task. Indeed, it ensures the uniformity of the structure of the lexical entries having the same grammatical category. Thus, the same acquisition models can be employed while ensuring the appropriate constraints. Moreover, it offers a finely structured model in a way that we can accede to each knowledge separately. Hence, appropriate set of acquisition constraints can be provided for each knowledge or relatinship.

In conclusion, we can state that LMF can be considered as a solution for the collaborative enrichment of dictionaries with a large linguistic coverage as it is already confirmed as a solution for modeling such dictionaries.

4   Proposed Collaborative Approach

4.1   Overview of the Approach

The proposed approach aims the enrichment of LMF-normalized dictionaries with a large linguistic coverage. It ensures their integrity by protecting the contents of such dictionaries given the fact that they are a reference sources for natural languages. Thus, because of the kind of the requested knowledge, the user should be an expert (i.e., linguist, lexicographer) in the matter. In addition, an automatic control mechanism is applied using appropriate rules in order to guarantee some required properties such as consistency, completeness and non-redundancy.
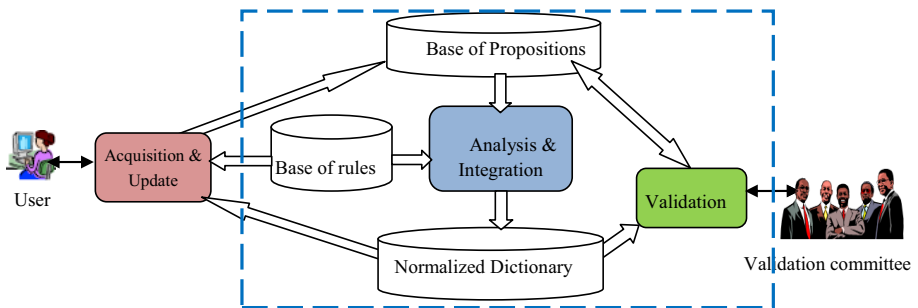

Fig. 3. Collaborative approach of the enrichment

As shown in the Figure 3, the enrichment process is based on three phases and uses apart the normalized dictionary, a base of rules and a base of user propositions. The enrichment concerns the adding of a new LE or the updating of an old one which will

be saved in the base of propositions. After that, we must launch the analysis and the integration of propositions which generate the normalized form of the LE and detect conflicts. These conflicts will be studied and resolved by the committee through the validation phase. All validated LEs we will be passed by the analysis before being recorded in the normalized dictionary.

4.2 Phases of the Approach

a. **Acquisition and Update.** This phase is composed by a set of sub-modules according to linguistic LMF extensions (i.e., morphological, syntactic, semantic) and requires the completeness rules.

The user can either introduce a new LE or update an old one. In the first case, a new LE must have at least one lemma and a POS. In the second one, the user seeks the existing LE and then modifies the desired knowledge and argues its proposal. The added or updated LE will be saved in the propositions base.

b. **Analysis and Integration.** This phase must be executed after adding or updating a LE. It is an automatic phase including the following four steps:

– Analysis: For each proposition, it checks whether it is an old or a new LE. In the case of a new LE, it passes to the second step. In the case of an old one, it looks at the kind of the update: a lack or an error.
– Allocation of identifiers: For each proposition of LE, it generates its ID afterward the ID of every sense.
– Integration: This step uses rules for ensuring completeness and non-redundancy. It saves all the new knowledge of a proposition in the dictionary except links (i.e., RelatedForm, SenseRelation, SemanticPredicate, SynSemCorrespondence).
– Establishing links: This step is based on the consistency rules and on the non-redundancy rules. For every proposition of LE, it ensures the links with the others LEs, relying on the rules of consistency. It starts by the morphological relation, then the Semantic Relation (SR).

c. **Validation .** This phase is provided by the validation committee. It treats case by case the conflict of users's propositions.

5   Control of the Enrichment

5.1 Completeness Rules

The rule 1 deals with the completeness of a new LE. It checks the mandatory knowledge.

Rule 1:   *if (New (LE) and LE = CF) Then   Mandatory (POS) and (Lemma = CF) EndIf*

Any new LE of a Canonical Form (CF) must have a POS and a Lemma to be stored is the D. *New()* verifies if the element received as parameter is new and *Mandatory()* controls the existence of mandatory knowledge.

Rule 2:   *if (New (S)) Then Mandatory (Def) EndIf*

Each new sense (*S*) must have at least one not empty definition (*Def*).

5.2   Non-redundancy Rules

These rules ensure the non-redundancy of LE in the dictionary.

Rule 3:   *if (New (SR)) Then   Single-SR (SiLE1, SjLE2) EndIf*

Any new SemanticRelation (*SR*) linking a current Sense *i* of *LE1* (*SiLE1*) and a Sense *j* of another entry *LE2* (*SjLE2*) must verify that there is no other *SR* linking *SiLE1* to *SjLE2*. *Single-SR()* verifies the uniqueness of the *SR*.

Rule 4 :   *if (New (SF) && (not Exist (SC, D))) Then   Create (SF)   EndIf*

 If the user proposes a new SyntacticFrame (*SF*) we must check if it already exists in the dictionary. *Create()* allows the creation of a new element. *Exist()* verifies the existence of the first parameter in the second one. *MorphS()* verifies if the LE is derived from the received stem. *Link()* establish the link between its parameters.

5.3   Consistency Rules

These rules deal with the links between a new LE and the other ones which are in the dictionary to ensure the consistency. We can classify them by their linguistic level.

- **At morphological level:**

Rule 5 : *If (Type (LE1, FD) && MorphS (LE1, S)) Then If (not Exist (S, D)) Then   Create (LE2, S)   EndIf*

Link (LE1, LE2) EndIf

If a LE1 is a Derived Form from a stem (S), we need to verify the existence of this stem in the dictionary. If S is not in the dictionary, we create a LE2 for this stem. Afterward, we provide the link between LE. *Type()* checks if the type of the first parameter is equale to the second one.

- **At semantic level:**

Every word has many senses and many synonyms; in the dictionary model we link the synonymy senses and not the lemmas. But, in the propositions base, we link the sense to the lemma which is not yet introduced in the base.

Rule 6 : *If (Exist-SR (SLE1, LE2))   Then   If   (not Exist (LE2, D)) Then New (LE2)   EndIf*

Link (SLE1, SLE2)   EndIf

If a sense of *LE1* has a Semantic Relation (*SR*) with *LE2*, we must verify the existence of *LE2* in the dictionary. After, we must ensure the link between the two senses: *SLE1* and *SLE2*.

6   Experimentation on an Arabic Normalized Dictionary

The choice of the experimentation case is justified, firstly by the fact that in our team we are working on the Arabic language, and secondly by the existence of a standardized model and a first version of a dictionary constructed according to LMF[3].

6.1   Implementation of the System

At the implementation level, we used the Flex development environment[4] to propose a fully customized Web2.0 application and especially authorizing a way of working offline (Data Synchronization is carried out automatically on reconnection) to minimize network traffic and upload.

6.2   Experimentation Results

At present, the prototype is hosted locally on the intranet of our laboratory. We designed four users which are experts in the lexicography domain to test the implemented system. They started with the diet of 10000 entries: 4000 verbs, 5980 names and 20 particles. Each user has to deal with the entries starting with a list of Arabic letters. We notice that they have had difficulty starting to discover the interfaces and the requested knowledge, which needs the development of a user guide to help new users. Moreover, the experts can work offline and they connect only once to send proposals.

   In order to evaluate the developed system, we conducted a qualitative assessment of the fragment introduced by human experts. Then we observed a few gaps at all levels of control, namely the completeness, the consistency and the non-redundancy. When analyzing the results, we noticed that some specific rules for the case of Arabic must be added. They are related to specific aspects of Arabic morphology that uses the concepts of root and schema pattern. These rules are then formulated and implemented. In addition, some problems related to the link establishment were noted, notably for semantic links. Indeed, when the user does not mention a sense of the word supporting its synonym, the system is unable to assure this kind of link although it exist when a human analyzes the entries.

   The role of the committee was limited to deal with the request of canceling some knowledge that was given on purpose to test the system. Indeed, this kind of update might cause incoherence in the content of the dictionary.

7   Discussion

The forcefulness of the Wiki approach as a remote collaborative tool has generated the appearance of several projects dealing with the collaborative construction of

---

[3] www.miracl.rnu.tn/Arabic-LMF-Dictionary
[4] www.adobe.com/fr/products/flex.html

lexical resources [7] [4] but the famous one is the Wiktionary interested in dictionaries' development of various languages [13]. The contribution in these projects is devoted for all kinds of users and the treated knowledge is not detailed. For example, the Wiktionary is an open dictionary and it is based on a simple pattern containing the part of speech, etymology, senses, definitions, synonyms and translations. However, it does not treat the syntax and it do not have the means to link synonyms and related senses. The simplicity of its structure makes the alimentation task within reach of no experts in this field and his contents suffer from a lack of knowledge and precision.

Furthermore, the use of an LMF standardized dictionary is a strong point of our project. The related model is complex and provides a wide linguistic coverage [2]. The robustness of the model makes the enrichment task unreachable by everyone, despite the GUI which facilitates the edition and the check of information in semi-structured documents. This task is controlled by a set of rules ensuring the properties of completeness, coherence and non-redundancy. Moreover, a validation committee resolve the conflicts of propositions to guarantee the consistency of the dictionary content.

8   Conclusion

An electronic dictionary rich in linguistic knowledge and lifelong up-to-date is highly requested. However, its construction is very difficult and expensive. Thus, we proposed a collaborative approach to the enrichment and the update for an LMF normalized dictionary. This approach uses a set of rules to ensure the properties of completeness, coherence and non-redundancy which are quite recommended because of the fine structure and the expressive content of the dictionary.

An enrichment system has been already implemented for Arabic and tested on a dictionary fragment containing about 10000 lexical entries. However, some improvements are planned such as covering new semantic links (other than synonym) or generating semantic links while analyzing sense knowledge of existing entries.

It is very important to proceed to a quantitative evaluation using appropriate metrics. Also, it is interesting to carry out more experiment on other languages to define the specific rules and the general ones. Finally, we consider putting up the system at the Web for a wide experiment.

References

1. Arregi, X., et al.: Semiautomatic of the EuskalHiztegia Basque Dictionary to a queryable electronicform. In: L'objet, LMO 2002, pp. 45–57 (August 2002)
2. Baccar, F., Khemakhem, A., Gargouri, B., Haddar, K., Ben Hamadou, A.: LMF standardized model for the editorial electronic dictionaries of Arabic. In: 5th International Workshop on Natural Language Processing and Cognitive Science, NLPCS 2008, Barcelone, Espagne, June 12-13 (2008)

3. Bellynck, V., Boitet, C.: and Kenwright J., Construction collaborative d'un lexique français-anglais technique dans IToldU: contribuer pour apprendre. In: 7èmes Journées scientifiques du réseau LTT (Lexicologie Terminologie Traduction) de l'AUF (agence universitaire de la francophonie), Bruxelles (2005)

4. Daoud, M., Daoud, D., Boitet, C.: Collaborative Construction of Arabic Lexical Resources. In: Proceedings of the International Conference on MEDAR 2009, Cairo, Egypt (2009)

5. Dendien, J., Pascal, M., Pierrel, J.-M.: Le Trésor de la Langue Française informatisé: Un exemple d'informatisation d'un dictionnaire de langue de référence. TAL 44(2), 11–39 (2003)

6. Francopoulo, G., George, M.: ISO/TC 37/SC 4 N453 (N330 Rev.16), Language resource management- Lexical markup framework, LMF (2008)

7. Garoufi, K., Zesch, T., Gurevych, I.: Representational Interoperability of Linguistic and Collaborative Knowledge Bases. In: Proceedings of KONVENS 2008 Workshop on Lexical Semantic and Ontological Resources Maintenance, Representation, and Standards, Berlin, Germany (2008)

8. Gurevych, I., Eckle-Kohler, J., Hartmann, S., Matuschek, M., Meyer, C., Wirth, C.: Uby - A Large-Scale Unified Lexical-Semantic Resource Based on LMF. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), pp. 580–590 (April 2012)

9. Khemakhem, A., Gargouri, B., Haddar, K., Ben Hamadou, A.: LMF: Lexical Markup Framework. In: LMF for Arabic, pp. 83–96. Wiley Editions (March 2013)

10. Khemakhem, A., Elleuch, I., Gargouri, B., Ben Hamadou, A.: Towards an automatic conversion approach of editorial Arabic dictionaries into LMF-ISO 24613 standardized model. In: Proceedings of the International Conference on MEDAR 2009, Cairo, Egypt (2009)

11. Mangeot, M., Enguehard, C.: Informatisation de dictionnaires langues africaines-français. Actes des Journées LTT 2011 (Septembre 15-16, 2011)

12. Mangeot, M., Sérasset, G., Lafourcade, M.: Construction collaborative d'une base lexicale multilingue, le projet Papillon. TAL 44(2), 151–176 (2003)

13. Sajous, F., Navarro, E., Gaume, B., Prévot, L., Chudy, Y.: Semi-Automatic Endogenous Enrichment of Collaboratively Constructed Lexical Resources: Piggybacking onto Wiktionary. In: Proceedings of the 7th International Conference on Natural Language Processing (IceTAL 2010), Reykjavik, Iceland (2010)

14. Sarkar, A.I., Pavel, D.S.H., Khan, M.: Collaborative Lexicon Development for Bangla. In: Proceedings International Conference on Computer Processing of Bangla (ICCPB 2006), Dhaka, Bangladesh (2006)

Enhancing Machine Learning Results for Semantic Relation Extraction

Ines Boujelben, Salma Jamoussi, and Abdelmajid Ben Hamadou

Miracl-Sfax University, Sfax-Tunisia
Technopole of Sfax: Av.Tunis Km 10 B.P. 242, Sfax-Tunisia
Boujelben_ines@yahoo.fr, jamoussi@gmail.com,
abdelmajid.benhamadou@isimsf.rnu.tn

Abstract. This paper describes a large scale method to extract semantic relations between named entities. It is characterized by a large number of relations and can be applied to various domains and languages. Our approach is based on rule mining from an Arabic corpus using lexical, semantic and numerical features.

Three primordial steps are needed: Firstly, we extract the learning features from annotated examples. Then, a set of rules are generated automatically using three learning algorithms which are Apriori, Tertius and the decision tree algorithm C4.5. Finally, we add a module of significant rules selection in which we use an automatic technique based on many experiments. We achieved satisfactory results when applied to our test corpus.

Keywords: Semantic relation, Named Entity, supervised learning, rules mining, rules selection.

1   Introduction

Relation extraction presents the task of discovering useful relationship between two Named Entities (NEs) from text contents. As this task is very useful for information extraction applications like business intelligence and event extraction as well as the natural language processing tasks such as question-answering, many research works have been already performed. Some works are rule-based which rely on hand-crafted patterns [3]. Others use machine learning algorithms to extract relations between NEs. We distinguish unsupervised machine learning methods [5, 11] that conduct to extract words between NEs and cluster them in order to produce many clusters of relations. Hence, considered relations must occur many times between NEs within the same sentence which is not always possible in Arabic sentences. However, semantic relations can occur either before the first NE, between NEs or after the second NE. Alternative supervised learning methods can be used to automatically extract relation patterns based on annotated corpus and linguistic features [7, 8, 14]. Inspired of this latter, we proposed our supervised method based on rules learning. The remainder of this paper is organized as follows: Firstly, we explain our proposed method. Then, we

present the evaluation results when our method is applied on a test corpus. Finally, some conclusions are drawn in order to structure future work.

2   Proposed Method

The main idea of our work is to generate automatically rules which are used to extract semantic relation that may occur between NEs. Indeed, a rule is defined as a set of dependency path indicating a semantic link between NEs.

Our method consists of three steps: the learning feature identification, the automatic generation of rules using machine learning algorithms and finally the selection of significant rules that aims to iteratively improve the performance of our system. First of all, we extract sentences that contain at least two NEs from our training corpus. As we know, the Arabic language is characterized by its complex structure and its long sentences [7]. For that, when analyzing our Arabic training sentences, we note that numerous relations are expressed through words surrounding the NEs that can be either before the first NE, between NEs or after the second NE. Additionally, some NEs are unrelated, in despite of their presence in the same sentence. Some previous works like [14] use a dependency path between NEs to estimate if there is a relation. To address this problem, we seek to limit the context of semantic relation extraction in order to guarantee the existence of relation between the attested NEs. Referring to linguistic experts and the study of examples, the clause segmentation task can present a better solution that can tackle this problem on average of 80%. This extraction required an Arabic clauses splitter [7] as well as an Arabic NEs [9] recognition tools. Arabic language suffers from the lack of available linguistic resources like annotated corpora and part-of-speech tagging. Therefore, we need to spend additional effort to label and verify our linguistic resources used for the learning. Hence, we need an efficient part-of-speech tagging to produce morphological tag of each context word or symbol (punctuation mark) given that Arabic is an agglutinative language in which the clitics are agglutinated to words. For example, the conjunction (و/and) or the pronoun (هم/them) can be affixed to a noun or a verb and thus causes several morphological ambiguities. We elaborated a sample transducer for surface morphological analysis using NooJ platform [12] based on Arabic resources of [9].

2.1   Learning Features Identification

Many early algorithms use a variety of linguistic information including lexical, semantic and syntactic information [13, 15]. Many others use no syntactic information like the DPIRE algorithm [4]. In our case, we focused only on lexical, numerical and semantic features without syntactic information. The features used are as follows:

Our method is distinct from previous works [2, 4] which aim to recognize the semantic class of relation. We extract the position of word surrounding the NEs that reflect the semantic relation. So, we are not limited to a defined number of classes.

Table 1. Identification of the learning features

Type	Feature	Description
Lexical	POS_W1_C1	The part of speech tag of the first word before the first NE.
	POS_W2_C1	The part of speech tag of the second word before the first NE.
	POS_W3_C1	The part of speech tag of the third word before the first NE.
	POS_W1_C2	The part of speech tag of the first word between the two NEs.
	POS_W2_C2	The part of speech tag of the second word between the two NEs.
	POS_W3_C2	The part of speech tag of the third word between the two NEs.
	POS_W1_C3	The part of speech tag of the first word after the second NE.
	POS_W2_C3	The part of speech tag of the second word after the second NE.
	POS_W3_C3	The part of speech tag of the third word after the second NE.
Semantic	NE1	The first named entity tag: PERS, LOC, ORG.
	NE2	The second named entity tag: PERS, LOC, ORG.
	Pair	The appearance order of NEs.
Numeric	NB_W_C1	Number of words before the first NE.
	NB_W_C2	Number of words between the two NEs.
	NB_W_C3	Number of words after the second NE.

From annotated clauses, we are able to build our training dataset by extracting our learning features. In fact, fifteen features are identified in which fourteen are retrieved automatically, and the last one, the position of relation, is manually identified. Our training data is composed of 1012 sentences collected from electronic and journalistic articles in Arabic. They contain 8762 tokens including word forms, digits, and delimiters. These sentences have 2000 NEs, in which only 1360 NEs are related.

2.2   Automatic Rules Generation

Actually, we aim to extract automatically rules with high precision. Therefore, two association rules algorithms were chosen for their high performance in terms of confidence and support. We utilized the standard algorithm of association rule induction Apriori [1] and the Tertius algorithms [6] which produce predictive and first-order rules. Here is a sample of rules produced by these algorithms.

Rule 1: *NE1=PERS and NE2=LOC and POS W1 C1=verb and POS_W1 _C2=nom and POS W2 C2=prep* ➔ *class=W1C2*

Rule 2: *NE1=PERS and NE2=PERS and POS_W1_C1=verb and NB_W_C2=0* ➔ *class=W1C1*

We investigated also the C4.5 algorithm [10] which produces classifiers expressed as decision trees. From this latter, we generate the classification rules structured like the association rules forms like shown in this example.

Rule3: *NB_W_C2 <= 1 and POS_W1_ C2=verb and NB_W_C2 <= 0 and POS_W1_C1=verb*➔ *class=W1C1*

2.3   Selection of Significant Rules

The major drawback of the learning algorithms is the large number of rules generated to cover all the superficial variations in clause constructions. These rules can be in some sense interesting or not. For this reason, we envisage to apply refinement operations on them in order to cover further instances of our training data set with higher precision. These obtained rules have to be filtered firstly in terms of size. That means

the number of attributes that compose one rule. In fact, we believe that each rule composed of one or two attributes will induce erroneous and redundant results. Hence, the rules composed of less than 3 attributes will be neglected. The second filtration consists in hiding the rules by tuning their confidence[1] and support[2]. The higher theses values, the more often the rule items are associated together. In our case, the rule getting a confidence value below a threshold value will be removed. Next, we opt to an enrichment step for our obtained rules by generating others new rules from the best ones. This enrichment has the advantage of increasing the coverage of our system. This means, for each rule disposing of a set of more than three attributes, we eliminate iteratively one attribute to obtain an equivalent number of derived rules. Next, these selected rules as well as the top rules are applied to our training data set. So, we will obtain a very large number of rules in which some of them are redundant.

We proceed then to a third step of selection in which we compare each target rule with its derived rules in order to satisfy two assumptions: If one of the derived rules holds with a confidence value more than a specified threshold and gets the highest support, then it will be selected. In the case that all derived rules have confidence values below the threshold value, we will conserve only the target rule and eliminate all its derived rules.

3   Experiments and Results

Two evaluation experiments are presented: the first one assesses the rules selection and the second evaluates the effectiveness of our proposed method.

3.1   Selection Rules Evaluation

The evaluation of significant rules selection when they are applied to our training data obtained the following results shown in table 2.

Table 2. The obtained results for each selection level

Selection rules levels	Rules number	Precision	Recall	F-score
All rules	470	60%	65.6%	62.57%
First level (attribute number >=3)	350	67%	58%	62.18%
Second level(Confidence>**0.6**)	170	75.4%	62.27%	68.21%
Third level(Enrichment)	638	72%	66.8%	69.3%
Fourth level (significant rules)	170	74.9%	66%	70.16%

These results demonstrate the amelioration of the F-measure values among the selection rules levels (from 62.5% to 70.1%). The best F-measure value is obtained in the final level which proves the effectiveness of our selection module. Thus, we succeed to pick the best compromise between these three parameters: precision, recall and rules number.

---

[1] The confidence shows how frequently the rule head occurs among all the groups containing the rule body.

[2] The support presents the number of instances in which the rule is applicable.

For the second selection level which consists of filtering the obtained rules by comparing them to a confidence value threshold, we have first to define the right value of this threshold. Therefore, we plotted the precision/recall curves (figure 1) by varying the confidence threshold value. As mentioned in Figure1, the threshold is fixed to 0.6.
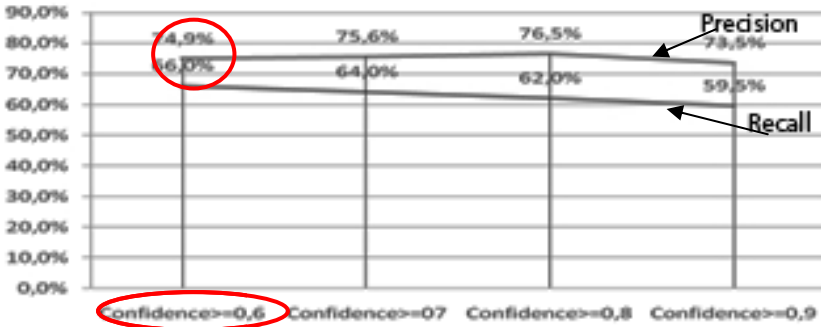

Fig. 1. Precision and recall curves function of the confidence threshold

3.2   Method Evaluation

We created a new Arabic corpus distinct from the training corpus. It is composed of electronic and journalistic articles collected from the web sites. It contains 553 texts and about 2800 text units or sentences, 53197 word forms and more than 20 000 different types of tokens. It contains 900 NEs in which 420 are related together with semantic relations. The application of the resulted significant rules ordered in terms of confidence and support on our test corpus gives the following results.

Table 3. Evaluation of our method

	Precision	Recall	F-score
Test corpus	70%	53.52%	60.65%

The results presented in table3 show that our relation extraction system is quite precise. However, it has a low recall, since it cannot handle exceptional relations between NEs. Indeed, our system is able to extract only explicit relations that are expressed through a special word or a punctuation mark in the sentence. Whereas implicit relations that are not indicated directly by specific words are difficult to be extracted since the output of our system is the word indicating the relation between NEs. The recall errors are also due to the influence of the NE recognition step. In effect, some Arabic NEs like the organization type entity has not been recognized which intricate the relation discovering. In the other hand, the difficulty to identify the right type of NE poses a problem in relation extraction, for instance "Tunis / تونس "could be either the name of a person or the name of a country. Thus, this problem can produce errors when applying a rule to the associated instance. So, to resolve this kind of problems, it is crucial to have a very efficient NE recognition tool for the Arabic language.

4   Conclusion and Perspectives

Our proposed method elaborates a set of two learning steps: The first one is used to generate automatically the rules through the combination of three learning algorithms. The second serves to select the significant rules from the generated ones. Unlike other recent works which are interested only in a specific domain, our method is general enough to be applied independently of both domain and language.

As perspectives, we still have other possible improvements to enhance the overall system performance. The addition of syntactic features and anaphora resolution can be used to improve the coverage of our system. Also, we tend to utilize human expertise rules with our learning selected rules to develop a hybrid approach which can improve our system capacities. Finally, it would be interesting to evaluate our system with other corpus in different languages and domains.

References

1. Agrawal, R., Srikant, R., Imielinski, T., Swami, A.: Mining Association rules between Sets of items in Large Databases. In: ACM, pp. 207–216 (1993)
2. Ben Abacha, A., Zweigenbaum, P.: A Hybrid Approach for the Extraction of Semantic Relations from MEDLINE Abstracts. In: Gelbukh, A. (ed.) CICLing 2011, Part II. LNCS, vol. 6609, pp. 139–150. Springer, Heidelberg (2011)
3. Boujelben, I., Jamoussi, S., BenHamadou, A.: Rule based approach for semantic relation extraction between Arabic named entities. In: NooJ (2012)
4. Brin, S.: Extracting patterns and relations from the World Wide Web. In: Atzeni, P., Mendelzon, A.O., Mecca, G. (eds.) WebDB 1998. LNCS, vol. 1590, pp. 172–183. Springer, Heidelberg (1999)
5. Culotta, A., Bekkerman, R., McCallum, A.: Extracting Social Networks and Contact Information from Email and the Web. In: CEAS (2004)
6. Flach, P.A., Lachiche, N.: The Tertius system (1999), http://www.cs.bris.ac.uk/Research/MachineLearning/Tertius
7. Keskes, I., Benamara, F., Belguith, L.: Clause-based Discourse Segmentation of Arabic Texts. In: LREC, pp. 21–27 (2012)
8. Kramdi, S.E., Haemmerl, O., Hernandez, N.: Approche générique pour l'extraction de relations à partir de textes. In: IC (2009)
9. Mesfar, S.: Analyse morpho-syntaxique automatique et reconnaissance des entités nommées en Arabe standard. University of Franche-Comté, Ecole doctorale langages, espaces, temps, socits (2008)
10. Quinlan, J.R.: Programs for Machine Learning. Morgan Kaufmann Publishers. Inc., San Mateo (1993)
11. Shinyama, Y., Sekine, S.: Preemptive information extraction using unrestricted relation discovery. In: HLT-NAACL, pp. 304–311 (2006)
12. Silberztein, M.: NooJ manual (2003), http://www.nooj4nlp.net
13. Stevenson, S., Swier, R.: Unsupervised Semantic Role Labeling. In: EMNLP, pp. 95–102 (2004)
14. Zelenko, D., Aone, C., Richardella, A.: Kernel Methods for Relation Extraction. JMLR, 1083–1106 (2003)
15. Zhou, G., Zhang, M., Donghong, J., Zhu, Q.: Tree kernel-based relation extraction with context-sensitive structured parse tree information. In: EMNLP-CoNLL (2007)

GenDesc: A Partial Generalization of Linguistic Features for Text Classification

Guillaume Tisserant, Violaine Prince, and Mathieu Roche

LIRMM, CNRS, Univ Montpellier 2
161 Rue Ada,
34090 Montpellier, France
{tisserant,prince,mroche}@lirmm.fr

Abstract. This paper presents an application that belongs to automatic classification of textual data by supervised learning algorithms. The aim is to study how a better textual data representation can improve the quality of classification. Considering that a word meaning depends on its context, we propose to use features that give important information about word contexts. We present a method named GenDesc, which generalizes (with POS tags) the least relevant words for the classification task.

1 Introduction

Textual data classification is an issue that has many applications, such as sentiment classification or thematic categorization of documents. This paper describes a classification method based on supervised learning algorithms. These algorithms require labelled data (i.e data composed of an input object and its class). They have a training phase during which they receive some features associated with the corresponding class label. After the training phase, the model built by the algorithm can associate a class to a set of features without label. The quality of classification depends not only on the quality of the learning algorithm, but also on data representation.

The usual method for textual data representation is the "bag of words" model: Each word is an input feature of a learning algorithm [1]. This representation considers each word as a separate entity, has the advantage of being simple and gives satisfactory results [2]. However, a lot of information is missed: For instance,the position of each word relatively to the others, is an important linguistic information that disappears with the "bag of words" representation. To store such an information, the n-gram model could be convenient. However, the multiplication of features and their lack of 'genericity' make this type of model unsatisfactory, since its *ad hoc* aspect impedes the quality of the learning algorithm output. For these reasons, it is necessary to find solutions to generalize features. In this paper, *generalizing features means replacing specific features, such as words by more generic features, such as their POS tags*, but is not restricted to this sole 'generalization'. This track has been quite well explored by

several other works in particular frameworks. [3] builds n-grams by replacing words by their POS-tag. [4] offers to consider only the bigrams in which the two words are related and to generalize the head word of the relationship. [5] uses canonical forms of words as a feature. This gives more general features, while remaining close to the concepts represented by the words, unlike POS-tags. These features can be used in n-grams . [6] builds n-grams based on canonical forms where some words (e.g nouns, verbs, adjectives, and adverbs) are replaced by their POS-tag. [7] uses n-grams of words **and** POS-tags to represent sentences. [8] demonstrates the value of using word sequences based on their syntactic relationship in sentences.

In this paper, we choose to provide a selective generalization (by POS tags) for a better classification of texts. Selective classification has been used in sentiment classification by [9], who built Noun-Adjective bigrams, where nouns were generalized (i.e. replaced by their POS tag): Adjectives are specific of feelings expression and are highly relevant to this particular type of classification. Our method, called GenDesc, is a procedure that generalizes those words that are the less relevant to a given task, without knowing the task in question. In a nutshell, it relies on distributions to decide whether a word is important or could be generalized.

In Section 2.1, we explain our issue and we present our goal. Section 2.2 develops the GenDesc method. In Section 3, experiments and results are described, and future work is presented in Section 4.

2   Towards an Appropriate Data Representation

2.1   Idea and Motivation

Using More General Features. The idea of replacing some words with their grammatical category comes from two observations: First, some words are more interesting than the others for textual data classification, and depending on the task, they should be used as features while others could be discarded. Secondly, all information cannot be transcribed in words only. Part of it is provided by the context: The sole occurrence of an important word is not enough by itself, it must be contextualized. For instance, the presence of a number of adverbs or adjectives may represent a crucial information to detect types of processed texts (e.g. opinion associated to a document or a sentence, etc.). In this paper we tackle the identification of words that can be generalized, because their are not discriminant. GenDesc is a method that will replace some words by their POS tags, according to a ranking function that evaluates the words frequency and their power of discrimination.

Word Position. Since words are not randomly inserted to make a sentence, their position, mostly in languages without declination, is a crucial information that is lost in the bag of words approach. Word position can be given by n-grams of words, which can be combined with generalization. So n-grams composed of

words, POS tags, or both, can be obtained, thus combining the initial Gen-Desc approach with the n-gram model. Each type of n-grams gives different information but each can be useful for the learning algorithm.

General Process. The approach we propose is divided into different steps. The first one determines the part-of-speech category of each word in the corpus. The next step selects the words that will be generalized. These will be used directly in their inflected form as features. The final step builds unigrams, bigrams, and trigrams from the remaining words and the generalized word labels. This data will form our features, used as a training input for the learning algorithm to build the prediction model for the classification task.

2.2 GenDesc: **Partial Generalization of Features**

Finding the words that are not so discriminant for a classification task, when the task is not known, and thus does not provide specific solutions, is the very issue tackled by GenDesc. To do so, we have tested some ranking functions which can assign a value to each word. If this value is smaller than a threshold, the word is generalized. The question is to determine the most accurate function, and a significative threshold. We have tested the TF (Term Frequency) function, the DF (Document Frequency) function (formula (1)), the IDF (Inverse Document Frequency) function (formula (2)), the D (Discriminence) function (formula (3)), and combination of these functions.

$$DF(x) = \text{number of documents which contain } x \qquad (1)$$

$$IDF(x) = \log \frac{\text{number of documents in the corpus}}{DF(x)} \qquad (2)$$

$$D(x) = \frac{\text{number of occurrences of word } x \text{ in the class which contains the most}}{\text{number of occurrences of word } x \text{ in the complete corpus}} \qquad (3)$$

3 Experiments

3.1 Experimental Protocol and Used Algorithm

Corpus. We have tested our method on a subset of the DEFT2007 corpus [10]. It is built with 28000 interventions of French congressmen, about legislation under consideration in the National Assembly. We worked on a subset of the corpus consisting of 1000 texts regarding the legislation, evenly balanced between sentences 'pro' and 'con'. We applied the SYGFRAN morphosyntactic parser [11] for French in order to obtain a POS tag to the words of the text. SYGFRAN recall and precision are quite high for French (more than 98% recall and precision for the DEFT corpus).

Learning Algorithms. We tested three different learning algorithms: Bayesian classification, decision trees, and SVM (Support Vector Machine), in the version implemented in Weka [12]. The Bayesian algorithm is NaiveBayes[1]. Decision trees algorithm is C4.5[2]. The SVM based algorithm is SMO (Sequential minimal optimization)[3]. Each algorithm is used with Weka default parameter. The results come from a 10-cross-validation, they are shown and commented in Section 3.2.

Ranking Functions. Each ranking function provides values in a different value domain. To allow a fair comparison, we have normalized all functions between 0 and 1.

3.2   Results

GENDESC. The different ranking functions in order to generalize words have been tested. The quality of obtained results are independent of the learning algorithm (see last paragraph of this section). Table 1 shows the results obtained using the NaivesBayes algorithm without the use of n-grams with different functions and thresholds. The use of POS tags (only) as features gives an accuracy at 53.75%. The use of words as features gives an accuracy at 60.41%. We consider these results as a baseline. Discriminence function (D) seems quite appropriate

Table 1. Table showing the accuracy according to different functions with different thresholds. Values in bold correspond to the values better than the baseline.

threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7
D	**63.49**	**65.11**	**68.26**	**64.60**	**63.18**	**61.26**	58.82
DF	59.23	58.73	58.72	58.82	58.92	59.33	59.74
IDF	59.26	54.87	54.56	55.17	53.75	54.67	54.67
TF	53.55	53.65	53.65	53.65	53.75	53.75	53.75
DFD	**60.95**	**60.85**	59.74	58.42	58.92	58.92	58.82
DIDF	**62.17**	60.41	53.85	53.85	53.85	53.85	53.85
TFD	**63.18**	**64.60**	**67.85**	**65.10**	**63.59**	60.24	59.84
TFIDF	55.48	54.57	54.57	54.67	53.65	53.65	53.55
TFDF	59.33	58.82	58.62	58.32	59.26	59.63	59.53

D: Discriminence , **DF**: Document Frequency, **IDF**: Inverse Document Frequency, **TF**: Term Frequency

for our purpose. Results show that when combined with TF (Term Frequency), its accuracy increases, while other functions degrade its performance The optimal threshold changes according to the function. If we consider the function D, the optimal threshold is always around 0.3, regardless of the learning algorithm, and whether n-grams are used or not.

---

[1] `http://weka.sourceforge.net/doc/weka/classifiers/bayes/NaiveBayes.html`

[2] `http://weka.sourceforge.net/doc/weka/classifiers/trees/J48.html`

[3] `http://weka.sourceforge.net/doc/weka/classifiers/functions/SMO.html`

GenDesc **Combination with *n*-grams.**   We conducted experiments on the same corpus, taking into account *n*-grams of words. Table 2 shows the results obtained with function D at its optimal threshold (i.e 0.3). The simultaneous use of bigrams and trigrams instead of unigrams tends to give a lower score than using unigrams alone. The use of bigrams and/or trigrams combined with the use of unigrams gives sometimes a better result than using unigrams alone. However, note that the difference is still less than 2%. An interesting point is that this slight improvement from *n*-grams is effective with *n*-grams built from features coming out of the partial generalization, but remains irrelevant when using plain *n*-grams of words. This suggests that the combination of *n*-grams and GenDesc is quite promising.

Machine Learning Algorithms.   Experiments with several learning algorithms were run, in order to compare their performance. Table 2 shows the obtained results. The first table shows GenDesc with a threshold at 0.3. The second one shows the use of *n*-grams of words. While NaiveBayes and SVM have relatively similar performance, the algorithm based on decision tree has lower performance, whether using words as features or those obtained with GenDesc. These results confirm that Unigram+Trigram gives always better results than the other combinations.

Table 2.   Tables showing the results obtained with the different learning algorithms

GenDesc:

n-grams	u	u + b	b	u + t	t	u+b+t	b + t
SVM	67.65	64.40	60.75	**68.46**	59.26	65.52	61.46
Bayes	68.26	67.55	62.78	**69.67**	58.11	68.36	61.66
Tree	59.84	59.74	55.88	**60.95**	52.23	60.85	55.68

Words:

n-grams	u	u + b	b	u + t	t	u+b+t	b + t
SVM	61.36	58.62	57.00	**62.98**	57.81	59.63	57.81
Bayes	60.14	60.24	59.26	**60.95**	57.99	60.55	59.53
Tree	55.38	54.77	52.13	**57.61**	54.16	52.43	55.38

u: unigrams, **b**: bigrams, **t**: trigrams

4   Conclusions and Future Work

In this paper, we proposed a representation of textual data that improves classification of document methods by generalizing some features (words) to their POS category when these words appear as less discriminant for the task. Our results show that this approach, called GenDesc is appropriate when classification is at stake, regardless from the nature of its criteria. Currently, we plan to construct partial generalization by building "*n*-grams" of words based on syntactic relations in place of *n*-grams of neighboring words. POS-tags, and POS-tags *n*-grams are more efficient with a more generic tag [9]. So we will try to obtain

more generic tag in our future work. Similarly, some words that are preserved can be replaced by their canonical form, in order to generalize information. Finally, we plan to use the method on another corpus, to test its appropriateness to other data and other languages.

References

1. Harris, Z.: Distributional structure. Word 10, 146–162 (1954)
2. Joachims, T.: Text categorization with suport vector machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
3. Gamon, M.: Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In: Proceedings of the 20th International Conference on Computational Linguistics, COLING 2004. Association for Computational Linguistics (2004)
4. Joshi, M., Penstein-Rosé, C.: Generalizing dependency features for opinion mining. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, ACLShort 2009, pp. 313–316. Association for Computational Linguistics (2009)
5. Porter, M.F.: Readings in information retrieval, 313–316. Morgan Kaufmann Publishers Inc. (1997)
6. Prabhakaran, V., Rambow, O., Diab, M.: Predicting overt display of power in written dialogs. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 518–522 (June 2012)
7. Kouloumpis, E., Wilson, T., Moore, J.: Twitter sentiment analysis: The good the bad and the omg! In: Adamic, L.A., Baeza-Yates, R.A., Counts, S. (eds.) ICWSM. AAAI Press (2011)
8. Matsumoto, S., Takamura, H., Okumura, M.: Sentiment classification using word sub-sequences and dependency sub-trees. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 301–311. Springer, Heidelberg (2005)
9. Xia, R., Zong, C.: Exploring the use of word relation features for sentiment classification. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING 2010, pp. 1336–1344. Association for Computational Linguistics (2010)
10. Grouin, C., Berthelin, J.B., Ayari, S.E., Heitz, T., Hurault-Plantet, M., Jardino, M., Khalis, Z., Lastes, M.: Présentation de deft 2007. In: Actes de l'atelier de clôture du 3eme Défi Fouille de Textes, pp. 1–8 (2007)
11. Chauché, J.: Un outil multidimensionnel de l'analyse du discours. In: Proceedings of the 10th International Conference on Computational Linguistics, COLING 1984, pp. 11–15. Association for Computational Linguistics (1984)
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11, 10–18 (2009)

Entangled Semantics

Diana Tanase and Epaminondas Kapetanios

University of Westminster, London, UK
diana.tanase@my.westminster.ac.uk, e.kapetanios@westminster.ac.uk

Abstract. In the context of monolingual and bilingual retrieval, Simple Knowledge Organisation System (SKOS) datasets can play a dual role as knowledge bases for semantic annotations and as language-independent resources for translation. With no existing track of formal evaluations of these aspects for datasets in SKOS format, we describe a case study on the usage of the Thesaurus for the Social Sciences in SKOS format for a retrieval setup based on the CLEF 2004-2006 Domain-Specific Track topics, documents and relevance assessments. Results showed a mixed picture with significant system-level improvements in terms of mean average precision in the bilingual runs. Our experiments set a new and improved baseline for using SKOS-based datasets with the GIRT collection and are an example of component-based evaluation.

1 Introduction

In 2009, the W3C announced a new standard: the Simple Knowledge Organisation System (SKOS) *a model for expressing the basic structure and content of concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies, folksonomies, and other similar types of controlled vocabulary*[1]. This meant that existing knowledge organization systems employed by libraries, museums, newspapers, government portals, and others could now be shared, reused, interlinked, or enriched. Since then, SKOS has seen growing acceptance in the Linked Data publishers community and more than 20% of existing Linked Open Data is using SKOS relations to describe some aspects of their datasets [1].

A relevant example for this article is the domain specific Thesaurus for the Social Sciences (TheSoz)[2] that has been released in SKOS format [7]. We use it to investigate its dual role as knowledge base for semantic annotations and as a language-independent resource for translation. For the experiments, we use the German Indexing and Retrieval Test database (GIRT) and a set of topics from the CLEF 2004-2006 Domain-Specific (DS) track. The focus of the experiments is to determine the value of using a SKOS resource in monolingual and bilingual retrieval, testing two techniques for annotation, explicit and implicit, and their effects on retrieval. Our results show a mixed picture with better results for bilingual runs, but worse average precision performance for the monolingual

---

[1] http://lod-cloud.net/state/
[2] http://datahub.io/dataset/gesis-thesoz

ones when compared to averages of all submitted runs for the corresponding CLEF DS track between 2004-2006.

This article is structured in four sections starting with related work in Section 2, a short description of key elements of SKOS in Section 3, semantic annotation experiments for monolingual and bilingual settings in Section 4, and our conclusions in Section 5.

2   Related Work

For almost twenty years now, the Semantic Web has been advocated as a space where *things* are assigned a well-defined meaning. In the case of text as the *thing*, a new technique has been developed for determining the meaning of the text and mapping it to a semantic model like a *thesaurus*, *ontology*, or other type of *knowledge base*. We refer to this as *semantic annotation* (SA) and adhere to the description specified in [3]: *Semantic annotation is a linking procedure, connecting an analysis of information objects (limited regions in a text) with a semantic model. The linking is intended to work towards an effective contribution to a task of interest to end users.*

The challenging aspect of semantic annotation is to process text at a deep level and detangle its meaning before mapping it to classes or objects from a formalized semantic model. Thesauri have previously been used for SA, for example in concept-based cross-language medical information retrieval performed using the Unified Medical Language System (UMLS) and Medical Subject Heading (MeSH) [5], yet with small impact. Among the reasons brought forward for this was the lack of coverage for a given document collection, the slow process of updating a static resource, and the problematic process of integration between different resources in different languages. The Semantic Web is set to change the approach by being a platform for interoperable, collaborative creation, dynamic (self-enriching) and distributed for language resources. Therefore, by taking a closer look at a interlinked thesauri formulated in SKOS we want to establish, if there is a positive impact on the overall retrieval.

3   Closeup on a SKOS Dataset

SKOS is a mechanism for describing concept schemes in a machine understandable way particularly aimed to be used by semantic technologies [1]. A *concept scheme* is a set of categories of knowledge at different granularity levels. This includes taxonomies, thesauri, and other vocabularies. SKOS itself does not provide solutions for how to create concept schemes, but how to represent them. The value of using SKOS resources is in the lightweight representation of domain specific vocabulary and categorizations.

Specifically, a SKOS description of a concept scheme contains a range of basic information about its concepts and the relations between them. Figure 1 shows a snapshot of the details captured by the Thesaurus for the Social Sciences[3]

---

[3] `http://lod.gesis.org/pubby/page/thesoz/concept/10034311`

(TheSoz) for *school*. For each concept in this dataset, the SKOS concept specification incorporates a set of multilingual lexical labels: the unique preferred term, a number of alternative terms, and additional documentation such as definitions and optional notes that describe the concept scheme's domain. TheSoz is also a Linked Open Data resource with interlinks to DBpedia[4](the extracted structured content from the Wikipedia project), the AGROVOC[5] thesaurus containing specific terms for agricultural digital goods, as well as STW Thesaurus for Economics[6].
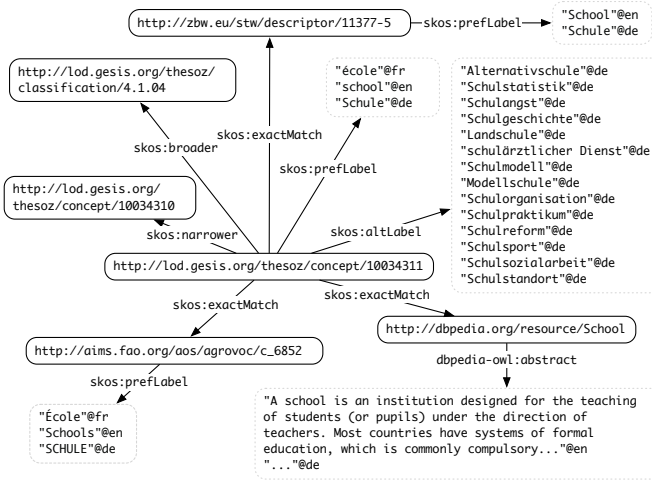
Fig. 1. TheSoz School Concept

In short, a SKOS resource has two levels of structure: a *conceptual level*, where concepts are identified and their interrelationships established; and a *terminological correspondence level*, where terms are associated (preferred or non-preferred) to their respective concepts.

4   Experiments

The experiments used 75 topics in English (EN) and German (DE) from the CLEF DS Tracks from 2004-2006 for searching the GIRT collection. This data is distributed by the European Language Association (ELRA)[7]. Previous results were ambivalent about the improvements possible through the use of domain specific resources in improving CLIR results and we wanted to set a new baseline and contrast it with previous work.

4.1   Development Setup

The following list describes the main components used in implementing and determining the results in the next section.

- Search Engine: Terrier 3.5 IR Platform[8]
- Knowledge Bases: TheSoz (DE,EN,FR), DBpedia, AGROVOC(19 languages), STW (mainly DE)
- Document Collection: GIRT consists of two parallel corpora in EN and DE, each with 151319 documents; documents are structured and can have a title, an abstract, and a set of thesaurus terms (on average 10 per document)
- Natural Language Processing: GATE[9] Embedded is an object-oriented framework for performing SAs tasks; APOLDA a GATE Plugin[10]
- Semantic repository: Virtuoso Universal Server[11]
- Translation Service: GoogleTranslate

4.2   Explicit Semantic Annotation – Finding Literal Occurrences of Concepts in a Text

The explicit semantic annotation aspect of our experiments relies on APOLDA (Automated Processing of Ontologies with Lexical Denotations for Annotation) Gate plugin [6] to determine annotations based on the SKOS-converted-to-OWL initial KOS resource. This plugin provides a scalable solution for basic text annotation where concepts have only a few labels. We annotated both the topic title and description, and added a new field *annotation* to all the topics. We did not disambiguate, since the topics and TheSoz are from the same subject domain. For example, for the Topic 175 with title *Parents' education level and children's school development* is annotated by the *education, information, parents* concepts.

4.3   Implicit Semantic Annotation – Beyond Literal Occurrences

In order to create a topic level annotation, which we will refer to as *keyconcept*, we used TheSoz's links to other SKOS datasets and for each of the concepts with an exact match from another scheme, we created a set of SPARQL queries that explored the other datasets looking for preferred and alternative labels. TheSoz concepts do not have definitions, but their exact DBpedia counterparts do. Thus, we extracted definitions for 5024 linked concepts out of the 8000 TheSoz specifies. The outcome is a set of textual signatures for each of the dataset's concepts. Some concepts have longer signatures than others. We built an index using Terrier over this set of textual signatures and used BM25 [2], as matching model to determine if a thesauri concept matches a certain topic. We took the first concept matching and added it to the *keyconcept* field of the topic together with any

---

[8] http://terrier.org/
[9] http://gate.ac.uk/download/
[10] http://apolda.sourceforge.net/
[11] http://virtuoso.openlinksw.com/

alternative labels. For Topic 172 a successful match was *parenting style*. Also, if a concept has several alternative labels they are grouped to be handled as synonyms by Terrier's query language. Thus, we rely on good precision P@1 and we set a threshold for the ranking score. Quality of SAs is hard to establish without a golden standard and in order to be transparent regarding the output of these two steps we are releasing the annotated topics and concept signatures we have built for TheSoz's concepts[12]. Note, that the translations for both types of annotations are performed using TheSoz's multilingual labels, while the topic's *title* is translated using Google's Translate service. We noticed that approximately 25% of topics the annotations are complementing each other and circumventing the topic's intent (e.g Topic 174: *Poverty and homelessness in cities* with implicit annotation *street urchin* and explicit annotations *homelessness* and *poverty*).

4.4   Results

After annotating and translating the topics, the necessary search indexes where built. We used language-specific stop-wordlists and stemmers and run a series of query formulations combinations, considering at turn pairings between the title (T), implicit annotations (A), and explicit annotations (C). We used PL2 (Poisson estimation for randomness)[13] as matching model and the default Query Expansion.

All results for Mean Average Precision (MAP) are listed in Table 1 and the percentage computations are performed against the second row of the table, which specifies the average MAP for past DS tracks. The best set of runs were obtained for the bilingual contexts in comparison to past experiments and Google Translate's web service clearly helps to achieve comparable performance, about 90%, to the monolingual runs. Yet, for combined runs using the topic's title and annotations, we saw an increase in performance relative to the average of all MAP values corresponding to that particular CLEF run. If we also compare across columns in Table 1, we notice that EN vs DE-EN is outperformed by the latter. Based on a human assessment of annotations for DE topics, and considering we did not use any word de-compounding tools, we noticed that there are a smaller number of annotations per topic 3-4 for DE topics as opposed to 5-6 for EN topics. This is evidence that when choosing the right annotation performance rises, but too many and of varied granularity lead to mixed results.

In one of the best runs at the CLEF DS Track 2005 that used TheSoz [4], mapping concepts to topics relied on inferred concept signatures based on the the co-occurrence of terms from titles and abstracts in documents and the concept terms associated with the document. This presupposes that the collection of documents has been annotated (this is true for GIRT) an assumption we found restrictive. Therefore, for the implicit annotations we used DBpedia descriptions that have allowed concepts too broad or too specific when matching a topic (e.g Topic 170: *Lean production in Japan* was matched to the *lean management* concept).

---

[12] `http://bit.ly/XUMrQK`

[13] `http://terrier.org/docs/v3.5/configure_retrieval.html#cite1`

Table 1. MAP for GIRT EN and DE Retrieval Results

DS 04-06	EN	DE	DE-EN	EN-DE
Average MAP for past runs	0.3960	0.3777	0.3127	0.2439
Query Formulation	**EN**	**DE**	**DE-EN**	**EN-DE**
T	0.3697 -6%	0.3617 -4%	0.3489 +11%	0.3308 +35%
C	0.1675	0.2096	0.1985	0.1339
A	0.2236	0.2158	0.2493	0.1611
T+C	0.3429	0.3183	0.3413 +9%	0.2671 +9%
T+A	0.3579	0.3440	0.3608 +15%	0.3136 +28%
C+A	0.2786	0.2327	0.2997	0.1858
T+C+A	0.3386	0.3033	0.3647 +16%	0.2687 +10%

5   Conclusions

The results in the previous section show the potential and some of the limitations of using the interlinked TheSoz as a knowledge base and language-independent resource for monolingual and bilingual IR settings. Though, the experimental results have not outperformed across the board, they set a new and improved baseline for using SKOS-based datasets with the GIRT DS collection, and are an example of component-based evaluation. Further work will concentrate on refining and extending the annotation process for the document collection and on experimenting with different levels of granularity for annotations in an IR context. We are aiming at generic solutions that are robust to the idiosyncrasies of interlinked SKOS datasets specifications.

References

1. SKOS Simple Knowledge Organization System Primer (February 2009), http://www.w3.org/TR/2009/NOTE-skos-primer-20090818/
2. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval the concepts and technology behind search, 2nd edn. Addison-Wesley (2011)
3. Kamps, J., Karlgren, J., Mika, P., Murdock, V.: Fifth workshop on exploiting semantic annotations in information retrieval: ESAIR 2012. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM 2012, pp. 2772–2773. ACM, New York (2012)
4. Petras, V.: How one word can make all the difference - using subject metadata for automatic query expansion and reformulation. In: Working Notes for the CLEF 2005 Workshop, Vienna, Austria, September 21-23 (2005)
5. Volk, M., Ripplinger, B., Vintar, S., Buitelaar, P., Raileanu, D., Sacaleanu, B.: Semantic annotation for concept-based cross-language medical information retrieval. International Journal of Medical Informatics 67(13), 97–112 (2002)
6. Wartena, C., Brussee, R., Gazendam, L., Huijsen, W.-O.: Apolda: A practical tool for semantic annotation. In: Proceedings of the 18th International Conference on Database and Expert Systems Applications, DEXA 2007, pp. 288–292. IEEE Computer Society, Washington, DC (2007)
7. Zapilko, B., Sure, Y.: Converting TheSoz to SKOS. Technical report, GESIS – Leibniz-Institut für Sozialwissenschaften, Bonn. GESIS-Technical Reports 2009|07 (2009)

Phrase Table Combination Deficiency Analyses in Pivot-Based SMT

Yiming Cui, Conghui Zhu, Xiaoning Zhu, Tiejun Zhao, and Dequan Zheng

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
{ymcui,chzhu,xnzhu,tjzhao,dqzheng}@mtlab.hit.edu.cn

Abstract. As the parallel corpus is not available all the time, pivot language was introduced to solve the parallel corpus sparseness in statistical machine translation. In this paper, we carried out several phrase-based SMT experiments, and analyzed the detailed reasons that caused the decline in translation performance. Experimental results indicated that both covering rate of phrase pairs and translation probability accuracy affect the quality of translation.

Keywords: Machine translation, Pivot method, Phrase table combination.

1   Introduction

In order to solve the parallel language data limitations, the pivot language method is introduced [1-3]. Pivot language becomes a bridge method between source and target languages, whose textual data are not largely available. When we choose a language as pivot language, it should provide a relatively large parallel corpus either in source-pivot direction, or in pivot-target direction.

In this paper, we focus on the phrase tables generated by two directions (*source-pivot, pivot-target*), that is *triangulation* method. This method multiplies corresponding translation probabilities and lexical weights in *source-pivot* and *pivot-target* phrase table to induce a new *source-target* phrase table.

2   Related Work

Utiyama and Isahara [3] investigate in the performance of three pivot methods. Cohn and Lapata [4] use multi-parallel corpora to alleviate the poor performance when using small training sets, but do not reveal the weak points of current phrase-based system when using a pivot method. What affects the pivot-based machine translation quality is discussed in general aspects by Michael Paul and Eiichiro Sumita [5], but not detailed explained in a certain aspect.

3   Pivot Method In SMT

When combining the two phrase tables generated by *source-pivot* and *pivot-target* corpora, we should take two elements into account.

The first element is phrase translation probability. We assume that source phrases are independent with target phrases. In this way, we can induce the phrase translation probability $\varphi(s\,|\,t)$ when given the pivot phrases as Eq.1.

$$\varphi(s\,|\,t) = \sum_p \varphi(s\,|\,p) \cdot \varphi(p\,|\,t) \tag{1}$$

Where s, p and t denotes the phrases in the source, pivot and target respectively.

The second element is lexical weight, that is word alignment information a and in a phrase pair (s,t) and lexical translation probability $w(s|t)$[6].

We assume a_1 and a_2 be the word alignment inside phrase pairs (s,p) and (p,t) respectively, and the word alignment a of phrase pair (s,t) can be got by Eq.2.

$$a = \{(s,t)\,|\,\exists p : (s,p) \in a_1 \,\&\, (p,t) \in a_2\} \tag{2}$$

Then we can estimate the lexical translation probability by induced word alignment information, as shown in Eq.3. In this way, we can use source-pivot and pivot-target phrase table to generate a new source-target phrase table.

$$w(s\,|\,t) = \frac{count(s,t)}{\sum_{s'} count(s',t)} \tag{3}$$

4   Experiments

In our experiments, the pivot language is chosen as English, because of its large availability of bilingual corpus. Our goal is to build a Chinese-Japanese machine translation system. The corpus is selected as HIT trilingual parallel corpus [7]. There are two ways to divide the corpus. The first is *parallel* one, which indicates that both directions share the same training sets; the second is *non-parallel* one, which means the training sets of two directions are independent with each other. The Statistics are shown in Table 1.

Table 1. Zh-en-jp parallel corpus

	Train	Dev	Test
Parallel	59733	2000	1000
Non-parallel	29800*2	2000	1000

4.1   Coverage of Phrase Pairs

The coverage of phrase pairs shows how many phrases appear in the phrase table, and it can be an indicator that reveals the difference between standard and pivot model.

The scales of each phrase tables are shown in Table 2. Then we extracted phrases separately from standard and pivot phrase table, and deleted all repeated phrases respectively. We can calculate the number of phrases. The results are shown in Table 3.

Table 2. The scale of two models

	Standard	Pivot
Parallel	1088394	252389200
Non-Parallel	1088394	92063889

Table 3. Number of phrases

	Parallel		Non-Parallel	
	zh	jp	zh	jp
Standard	521709	558819	521709	558819
Pivot	320409	380929	97860	131682

In general, we can see some problems revealed in figures above. Firstly, though pivot phrase table may be larger than the standard one in size (230 times bigger), the actual phrases are less than the standard one (about 60%). This reminds us that during the phrase table combination, some phrases would be lost. That is to say, the pivot language cannot bridge the phrase pairs in *source-pivot* and *pivot-target* directions. Secondly, due to a larger scale in phrase table and lower useful phrases, pivot phrase table brings so much noise during the combination. This would be a barrier, because the noise would affect both the quality and the efficiency in the translation process.

Then we carried out the following experiments to show what caused low phrase coverage. We extracted the phrase pairs (*s,t*) that exist in standard model but not in pivot model. When given phrase *s*, we searched the Chinese-English phrase table to get its translation *e*, and use corresponding phrase *t* to search the English-Japanese phrase table to get its origin *e'*. Then we compared output *e* and *e'*, and see what reasons that caused the failure in connecting phrases in two models. We calculated the number of phrase pairs that was successfully connected by pivot in the Table 5.

Table 4. Connected phrase pairs in pivot model

	Parallel	Non-parallel
Connected phrase pairs	310439(34.75%)	73044(9.84%)

As we can see above, in parallel models there are only 34.75% phrase pairs connected, and in non-parallel situation, the rate goes down to 9.84%. So we examined the output file, and noticed some phenomenon which accounts for low number of connected phrase pairs. Firstly, Arabic numerals can be converted into English (e.g. 100 -> one hundred); secondly, the word with similar meanings can be converted (e.g. 8.76% -> 8.76 percent); thirdly, punctuations can be removed or added (e.g. over -> over.).

4.2 Translation Probability Accuracy

We also investigated whether translation probability accuracy affects the translation result a lot. We found the intersection phrase pairs in standard and pivot phrase tables, and generated two new phrase tables, using the common phrase pairs of standard and

pivot phrase tables, and the probabilities of each. In this way, we can see in the condition of the same phrase pairs, how results differ when using different translation probability. The results are shown in Table 6, which the parameters were not tuned.

Table 5. BLEU scores of old and new generated models(with *parallel* data)

	Standard	Pivot
Old	26.88	17.56
New	24.99	21.44

We can see that, in new models, the variety of the probability brings a 3.55 BLEU score gap. We found a quite unusual phenomenon that, though new pivot model reduce to 0.85% of its original size, the BLEU score rise up to 21.44. This can also be a proof that there are too much noise in pivot phrase table. The noise affected the translation quality, and translation effectiveness is also impacted due to its large size.

5   Conclusion

The experiments showed that the translation result may decrease along with the change of coverage of phrase pairs and translation probability accuracy. We still need to improve the covering rate of phrase pairs, and we also should improve our translation probability accuracy, not merely using a multiplication of each probabilities.

References

1. de Gispert, A., Marino, J.B.: Catalan-English statistical machine translation without parallel corpus: bridging through Spanish. In: Proceedings of 5th International Conference on Language Resources and Evaluation, pp. 65–68 (2006)
2. Wu, H., Wang, H.: Pivot Language Approach for Phrase-Based Statistical Machine Translation. In: Proceedings of 45th Annual Meeting of ACL, pp. 856–863 (2007)
3. Utiyama, M., Isahara, H.: A comparison of pivot methods for phrase-based sta-tistical machine translation. In: Proceedings of HLT, pp. 484–491 (2007)
4. Cohn, T., Lapata, M.: Machine Translation by Triangulation: Making Effective Use of Multi-Parallel Corpora. In: Proceedings of the 45th ACL, pp. 348–355 (2007)
5. Paul, M., Sumita, E.: Translation Quality Indicators for Pivot-based Statistical MT. In: Proceedings of 5th IJCNLP, pp. 811–818 (2011)
6. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pp. 127–133 (2003)
7. Yang, M., Jiang, H., Zhao, T., Li, S.: Construct Trilingual Parallel Corpus on Demand. In: Huo, Q., Ma, B., Chng, E.-S., Li, H. (eds.) ISCSLP 2006. LNCS (LNAI), vol. 4274, pp. 760–767. Springer, Heidelberg (2006)

Analysing Customers Sentiments: An Approach to Opinion Mining and Classification of Online Hotel Reviews

Juan Sixto, Aitor Almeida, and Diego López-de-Ipiña

DeustoTech−Deusto Institute of Technology, Universidad de Deusto, Avenida de las Universidades 24, 48007, Bilbao, Spain
{jsixto,aitor.almeida,dipina}@deusto.es

Abstract. Customer opinion holds a very important place in products and service business, especially for companies and potential customers. In the last years, opinions have become yet more important due to global Internet usage as opinions pool. Unfortunately , looking through customer reviews and extracting information to improve their service is a difficult work due to the large number of existing reviews. In this work we present a system designed to mine client opinions, classify them as positive or negative, and classify them according to the hotel features they belong to. To obtain this classification we use a machine learning classifier, reinforced with lexical resources to extract polarity and a specialized hotel features taxonomy.

1 Introduction

As online purchases are becoming more common, the number of opinions we can find on the Internet grows rapidly. This implies that the Internet is a never ending source of feedback and information about what customers like and dislike about products. Additionally, the spread of web portals (e.g. Epinions.com) that allow sharing opinions about products and services, have prompted to many users to make use of widespread opinions before purchasing a product or contract a service. When used in the analysis of opinions, such as the automated interpretation of on-line reviews, semantic orientation can be extremely helpful in marketing, measures of popularity and success, and compiling reviews [1].

This paper focuses on the problem of opinion mining, particularly applied to hotel reviews and aims to develop a system that is capable of extracting and classifying different parts automatically, based on the sentiments polarity and specialized features. Our objective in this project is to be capable of analysing reviews, mining customers general opinion and dividing it into its components, in order to evaluate them separately depending on the feature referring to. In this way, we expect to achieve a system that composes commercial reports that make hotels aware of the strengths and weaknesses of their service. Opinion analysis is one of the most studied tasks related to Data Mining and Natural Language Processing (NLP) in recent years. Dave, Lawrence and Pennock [2] worked with an

Amazon.com corpus and assigned scores with different classification techniques. RevMiner [3] is a smartphone interface that utilizes Natural Language Processing techniques to analyze and navigate reviews, based on attribute-value pairs extraction. Other approaches have based their classification on identifying features and opinion keywords in a sentence. They obtained some feature-opinion pairs and generated a most frequent pairs summary [4].Similarly, Agerri and Garcia-Serrano [5] have created Q-Wordnet, a lexical resource consisting of WordNet[1] senses, automatically annotated by positive and negative polarity, especially designed for opinion mining applications. Despite Q-Wordnet taking WordNet as a starting point and focusing on classifying word senses by polarity, it was created to effectively maximize the linguistic information contained in WordNet using human annotators instead of applying supervised classifiers. Thanks to Q-Wordnet, ratings can be used as feature for word sense classification.

2   System Framework

Using machine learning and natural language processing (NLP) techniques, we have created a classifier to positively or negatively evaluate reviews. We have implemented a general purpose classifier that takes a review as input data and assigns to it a category label. Next, the same classifier evaluates the split up parts of the review. Currently, numerous works for sentiment analysis and text classification are developed using Maximum Entropy Model (MaxEnt) and Naïve Bayes Model and based in this works we have used the Stanford NLP Classifier[2] tool for building the maximum entropy models. The text classification approach involves labelled instances of texts for the supervised classification task, therefore the proposed framework, described in Figure 1, integrates multiple text analysis techniques and cover the whole text process task. This framework contains some separated components based on text mining techniques, expert knowledge, examples, lexico-semantic patterns and empiric observations, using a modular structure that allows to enable and disable components to adapt the software composition to achieve the maximum successful results.

When processing the text, we analyse each sentence separately to facilitate the process of filtration into categories. In the first step, we tokenize the sentences to analyse the words independently. During this step, we clean the text of irrelevant symbols and English stop words[3]. Then we use the Stanford lemmatization and part-of-speech (POS) tagger to use the lemmas, their POS tags and their raw frequency as classifier features.These components provide a classifier features candidate list for polarity evaluation. Lastly, we annotate words using Q-Wordnet to establish the polarity and determine if a text (word, sentence or document) is associated with a positive or negative opinion.

Using part-of-speech (POS) tagger information, we are able to relate an adjective with the noun it refers to, generating bigrams with a polarity assigned

---

[1] http://wordnet.princeton.edu/
[2] http://nlp.stanford.edu/software/classifier.shtml
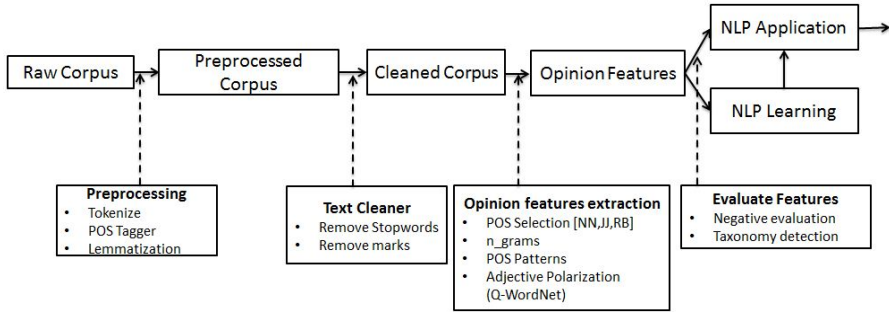[3] http://www.textfixer.com/resources/common-english-words.txt

Fig. 1. Structure of proposed framework

by Q-Wordnet. If a noun has several adjectives, these are evaluated separately. We added a filter that looks for negation words in the context of a sentence, to analyse if reversing bigrams polarization is necessary. Our system uses a classification process that evaluates the opinion polarization of a word in a context of a sentence, and link them with a hotel feature. If the word is a noun, system look for sentiment qualifiers in their context and contrast their polarity using Q-Wordnet and assigning a value. Next, evaluates the context and reverses the value when context is negative.

3 Experimental Results

The detailed experimental results are presented in Table 1. Experiments have been realized with the 1000 reviews corpus with two labels classification. We have used a 4-fold random cross-validation. The accuracy of the framework has been evaluated by comparing the classifier result with the manually labelled rating of reviews. We have found that the addition of POS Tagging to the framework have not provided a significant improvement, in fact, accuracy has decreased when we have used it without polarity detection . We have hypothesized this is due to word-category disambiguation is not relevant when the system only uses uni grams. In spite of this, POS Tagging has been also integrated in other system tasks, for example, during the word classifying process we have used the word category. The addition of polarity and negative tokens has improved the classifier significantly in relation to the base classifier, but we deem that this process is more relevant yet during the splitting task, due to the text to be analysed being shorter and the sentiment about concrete features being more important than the general sentiment of the review.

4 Conclusions

We have presented a a system for extracting opinions from reviews, focused on knowing the customers evaluation about the different features of hotel. Unlike

Table 1. Experimental results on review classification

Components	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
Base classifier	79.8	79.6	78.6	79.9
Base + POS	79.5	79.4	78.5	78.7
Base + ClearText + POS	80.0	79.8	78.5	78.9
Base + ClearText	80.5	80.6	78.8	79.3
Base + ClearText + Polarity	83.0	82.8	81.0	81.6
Base + ClearText + Polarity + POS	83.2	82.7	81.9	82.2
Base + Polarity + Negative	82.9	80.6	79.4	79.6
Base + ClearText + Polarity + Negative	**85.1**	**85.7**	**83.6**	**84.0**

other similar projects, we put an emphasis on dividing reviews among characteristics and classifying them separately. Also we have succeeded in using Q-Wordnet resources in combination with a probabilistic classifier. Even though our classifier does not improve significantly the precision of previous works, it becomes highly accurate achieving our ultimate objective; identify strengths and weaknesses of hotels based on clients opinions. The system is particularly accurate using a reviews set of the same hotel, when incorrect results are usually insignificant in relation with the high number of correct results.

References

1. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational Linguistics (2011)
2. Dave, K., Lawrence, S., Pennock, D.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th International Conference on World Wide Web, WWW 2003 (2003)
3. Huang, J., Etzioni, O., Zettlemoyer, L., Clark, K., Lee, C.: RevMiner: An Extractive Interface for Navigating Reviews on a Smartphone. In: Proceedings of the 25th ACM Symposium on User Interface Software and Technology (2012)
4. Zhuang, L., Jing, F., Zhu, X.-Y., Zhang, L.: Movie review mining and summarization. In: Proceedings of the ACM SIGIR Conference on Information and Knowledge Management, CIKM (2006)
5. Agerri, R., García-Serrano, A.: Q-WordNet: Extracting Polarity from WordNet Senses. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation (2010)

An Improved Discriminative Category Matching in Relation Identification

Yongliang Sun, Jing Yang, and Xin Lin[*]

Department of Computer Science and Technology, East China Normal University, China
`ylsun@ica.stc.sh.cn`, `{jyang,xlin}@cs.ecnu.edu.cn`

Abstract. This paper describes an improved method for relation identification, which is the last step of unsupervised relation extraction. Similar entity pairs maybe grouped into the same cluster. It is also important to select a key word to describe the relation accurately. Therefore, an improved DF feature selection method is employed to rearrange low-frequency entity pairs' features in order to get a feature set for each cluster. Then we used an improved Discriminative Category Matching (DCM) method to select typical and discriminative words for entity pairs' relation. Our experimental results show that Improved DCM method is better than the original DCM method in relation identification.

Keywords: Unsupervised Relation Extraction, Improved DF, Low-frequency entity pair, Improved DCM.

1 Introduction

Along with rapid growth of the digital resources and World Wide Web text information, corpus is becoming large and heterogeneous. More and more unsupervised information extraction (URE) methods are developed. URE aims at finding relations between entity pairs without any prior knowledge. URE is firstly introduced by Hasegawa [1] with the steps: (1) tagging named entities (2) getting co-occurrence named entities pairs and their context (3) measuring context similarities (4) making clusters of named entity pairs (5) labeling each cluster of named entity pairs. Rosenfeld and Feldman [3] compare several feature extraction methods and clustering algorithms and let identification of relations for further extraction by semi-supervised systems. Wang [4] applied URE in Chinese corpus and gives an overall summary and made an improvement based on the heuristic rules and co-kmeans method. In the relation identification step, Hasegawa [1] uses one most frequent word to represent the relation of a cluster. Chen [2] employs a discriminative category matching (DCM) to find typical and discriminative words to label for clusters not for entity pairs. Yan [5] use an entropy-based algorithm to ranking features.

For selecting a good relation word, we employs the improved DF [8] method to rank low-frequency entity pairs' features in this paper. Then we select a set of features for each cluster. Last, we propose an improved DCM method to select features to describe the relations. Experiments show that our method can select more accurate features to identify relations.

---

[*] Corresponding author.

2   An Improved Discriminative Category Matching

This paper uses ICTCLAS 2011 to segment sentences and we extract all nouns be-tween the two entities and two in the two sides by removing stopwords. Represent as $P_i = \{e_{i1}, e_{i2}, (w_1, t_1), (w_2, t_2), \ldots\}$. In k-means cluster method, we determine the range of clus-ter number k according [6] and use Silhouette index [7] as the evaluation metric to obtain the optimal value of k. The result is $\{C_1, C_2, C_3, \cdots, C_k\}$. To re-rank the fea-tures by their importance on relation identification, we use an improved DF method [8] with a threshold θ which is used to roughly distinguish the frequency of w_i. Table 1 is an example of function $f(w)$ and the importance can be calculated by Equation (2) [9]. Then we get a new whole features' order: $W = \{w_1', w_2', \ldots, w_m'\}$. For low-frequency entity pairs whose frequency is less than θ, we rearrange its order and get a new feature sequence $P_i = \{e_{i1}, e_{i2}, (w_1', t_1'), (w_2', t_2'), \ldots\}$. The detail of the Improved DCM me-thod is shown in Table 2.

Table 1. Features' information in each entity pair

	w1	w2	w3	…
P1	1	1	0	…
P2	1	0	2	…
…	…	…	…	…

$$f(w) = \begin{cases} 0, & w \ dosen't \ occur \ in \ P \\ 1, & w \ occurs \ lessthan \ \theta \ in \ P \\ 2, & w \ occurs \ more \ than \ \theta \ in \ P \end{cases} \quad (1) \qquad\qquad Weight = \sum_{0 \le i < j \le 2} n_i * n_j \quad (2)$$

Table 2. Improved DCM

Algorithm: Improved DCM

Input: $P = \{P_1, P_2, \ldots, P_n\}$ (all entity pairs), $C = \{C_1, C_2, \ldots, C_k\}$ (Cluster result)

Output: $\Pr = \{(P_1, r_1), (P_2, r_2), \ldots, (P_n, r_n)\}$ (r is the relation word that we extracted)

---

1. Begin
2. **For** each $P_j \in C_i$,
3.   Select the most important feature to get a cluster feature set: $F_{C_i} = \{w_1, w_2, w_3, \ldots\}$.
4. Use equation (3) to get the most important feature w_i in F_{C_i}.
5. **For** each $P_j \in C_i$,
6.   **If** $w_i \in P_j$, $r_j = w_i$;
7.   **Else** $r_j = w_1$; w_1 is the most important in P_j.
8. End

$$C_{i,k} = \frac{\log_2(df_{i,k}+1)}{\log_2(N_i+1)} \tag{3}$$

, where k means the kth feature(w_k) in cluster i. $df_{i,k}$ is the number of entity pair which contains w_k in cluster i. N_i is the number of entity pair in cluster i.

3   Experiments

In this paper, we ignore the Named Entity (NE) identification and suppose that we have got the entity pairs in corpus. Our method is tested on two different corpora. One of them is one month quantity of People's Daily (labeled by "PD") in 1998, including 334 distinct entity pairs and 19 relation words. The second is web corpus which we use Baidu search engine to get co-occurrence sentences (labeled by "Baidu"), which contains 343 distinct entity pairs and 8 relation words. Table 2 gives some details of these two datasets.

Table 3. Ralations in two corpora

PD		Baidu	
Relations	Number of Entity pairs	Relations	Number of Entity pairs
主席(president)	56	总统(president)	98
书记(secretary)	31	市长(mayor)	45
首相(premier)	7	首都(capital)	22
…	…	…	…

In order to measure results automatically, each entity pair's relation is labeled manually as Table 2 shows. Precision can be defined as follows:

$$P = \frac{N_{correct}}{N_{correct} + N_{error}} \tag{4}$$

$N_{correct}$ and N_{error} are the numbers of right and error results of relation extraction.

According to the evaluation metric in this paper, we can get the final precision of two different methods in our two dataset as Table 4 shows.

Table 4. The precision of different methods in two corpora

	PD	Baidu
DCM	86.60%	86.58%
Improved DCM	91.07%	95.91%

4   Conclusion and Future Work

We proposed an improved DCM method in Chinese corpus. We didn't treat low-frequent entity pairs as other methods which ignored them or regarded them as garbage. We combine feature selecting and improved DCM method to get a better relation word for each entity pair in relation identification. It solves the problem that similar relations are clustered into one group. It don't need to rely on high-precision clustering result.

In the future research we plan to find more accuracy relation words to describe the entity pairs' relation like some vice professions. One-to-many relationship is also waited to be researched.

References

1. Hasegawa, T., Sekine, S., Grishman, R.: Discovering Relations among Named Entities from Large Corpora. In: ACL 2004 (2004)
2. Chen, J., Ji, D., Tan, C.L., Niu, Z.: Unsupervised Feature Selection for Relation Extraction. In: IJCNLP 2005, JejuIsland, Korea (2005)
3. Benjamin, R., Ronen, F.: Clustering for Unsupervised Relation Identification. In: Proceedings of CIKM 2007 (2007)
4. Wang, J.: Research on Unsupervised Chinese Entity Relation Extraction Method, East China Normal University (2012)
5. Yan, Y., Naoaki, O., Yutaka, M., Yang, Z., Mitsuru, I.: Unsupervised relation extraction by mining Wikipedia texts using information from the web. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Suntec, Singapore, August 2-7, vol. 2 (2009)
6. Zhou, S., Xu, Z., Xu, T.: New method for determining optimal number of clusters in K-means clustering algorithm. Computer Engineering and Applications 46(16), 27–31 (2010)
7. Dudoit, S., Fridlyand, J.: A prediction-based resampling method for estimating the number of clusters in a dataset. Genome Biology 3(7), 1–21 (2002)
8. Xu, Y., LI, J., Wang, B., Sun, C.: A study of Feature Selection for Text Categorization Base on Term Frequency. In: Chinese Information Processing Front Progress China Chinese Information Society 25th Anniversary of Academic Conference Proceedings (2006)
9. Xu, Y., Huai, J., Wang, Z.: Reduction Algorithm Based on Discernibility and Its Applications. Chinese Journal of Computers 26(1) (January 2003)

Extracting Fine-Grained Entities
Based on Coordinate Graph

Qing Yang[1], Peng Jiang[2], Chunxia Zhang[3], and Zhendong Niu[1]

[1] School of Computer Science, Beijing Institute of Technology
[2] HP Labs China
[3] School of Software, Beijing Institute of Technology
{yangqing2005,cxzhang,zniu}@bit.edu.cn,
pengj@hp.com

Abstract. Most previous entity extraction studies focus on a small set of coarse-grained classes, such as person etc. However, the distribution of entities within query logs of search engine indicates that users are more interested in a wider range of fine-grained entities, such as GRAMMY winner and Ivy League member etc. In this paper, we present a semi-supervised method to extract fine-grained entities from an open-domain corpus. We build a graph based on entities in **coordinate list**s, which are html nodes with the same tag path of the DOM trees. Then class labels are propagated over the graph from known entities to unknowns. Experiments on a large corpus from ClueWeb09a dataset show that our proposed approach achieves the promising results.

Keywords: Fine-Grained Entity Extraction, Coordinate Graph, Label Propagation.

1   Introduction

In recent years, the task of harvesting entities from the Web excites interest in the field of information extraction. Most previous work focused on how to extract entities for a small set of coarse-grained classes. However, the distribution of entities within query logs of search engine indicates that users are more interested in a wider range of fine-grained entities [1], such as *Ivy League member* etc. In this paper, we introduce and study a task to serve the growing interest: web fined-grained entity extraction. Compared to traditional entity extraction, the main challenges in fine-grained entity extraction lie in 1) there are so many fine-grained classes defined
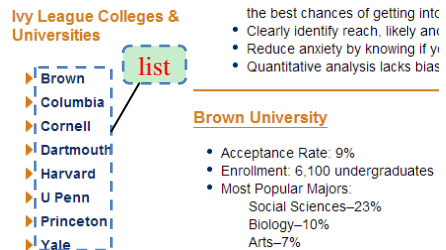
Fig. 1. Coordinate List of entities

---

[3] Corresponding author.

according to different tasks. 2) There is usually no context available. But the website editors usually use coordinate components such as table or list to facilitate readers to identify the similar content or concepts as Fig.1.Those entities in the same list or the same column of a table tend to belong to the same conceptual classes. We can build a graph to capture such co-occurrence relationship of entities. Then we propagate class labels of known entities to unknowns. Results demonstrate promising results .

The rest of the paper is organized as follows: Section 2 describes our approach to the task of fine-grained entity extraction. Evaluation is presented in Section 3. In Section 4, we discuss related work. Section 5 concludes the paper and discusses future work.

2   Entity Extraction for Fine-Grained Classes

In the task of fine-grained entity extraction, we are given 1) a set of coordinate lists $L = \{l_1, l_2, \ldots, l_n\}$ extracted from web pages, and 2) a list of fine-grained classes $C = \{c_1, c_2, \ldots, c_m\}$ defined in Wikipedia categories. We aim at inferring the classes that these entities belong to.

2.1   Coordinate Graph Construction

According to the observation that web site editors usually express similar contents with the same html tags, we assume all entities in the same coordinate list have similar classes. HTML list and table are two special cases of coordinate list.

First we group entities according to their text nodes' tag paths rooted from <HTML> into coordinate lists with the following rules: (1) the count of tokens (at least 2 and most 50); (2) non-characters (starting, ending or all); (3) some special types (e.g., number, date, URL, etc.) are filtered out; (4) the count of entities in a coordinate lists (at least 5). Then according to the co-occurrence of entities in different coordinate lists, we can build a **coordinate graph** $G = (V, E, W)$. V is a entity set $\{v_1, \ldots, v_n\}$. $e_{ij} \in E$ is a edge of entities v_i and v_j that co-occur in lists. $w_{ij} \in W$ reflects the similarity of class labels between two entities. In this paper, we use frequency of co-occurrence w_{co} and PMI w_{pmi} in different coordinate lists to measure the similarity between two entities. The PMI $w_{pmi}(e_{ij})$ can be computed as follows:

$$w_{pmi}\left(e_{ij}\right) = \log \frac{w_{co}\left(v_i, v_j\right) \times |C|}{f\left(v_i\right) \times f\left(v_j\right)} \tag{1}$$

where $w_{co}(v_i, v_j)$ is the frequency of co-occurrence of entity v_i and v_j. $f(v_i)$ is the frequency of entity v_i occurring in all coordinate lists. $|C|$ is the size of coordinate list set. Finally, we can normalized w_{co} and w_{pmi} to build the similarity matrix W.

2.2   Class Label Propagation on Coordinate Graph

In this paper, we use Wiki3C [2] to map known entities to Wikipedia articles and get the corresponding classes. Given all entities $V = \{v_1, v_2, \ldots, v_l, v_{l+1}, \ldots, v_n\}$ appear in coordinate lists, we assume the first l entities can be mapped into Wikipedia. Then, we can propagate the class labels of first l entities to the remaining unknown entities v_u $(l+1 \le v_u \le n)$ over the coordinate graph. Let $C = \{c_1, c_2, \ldots, c_m\}$ are the fine-grained classes extracted from Wikipedia. We assign a label tor $\boldsymbol{f}_i = \{p_{i1}, p_{i2}, \ldots, p_{im}\}$ for each entity v_i in the coordinate graph to represent the probability of each class that an entity belongs to. Initially, p_{ij} is assigned as follows:

$$p_{ij} = \begin{cases} 1 & \text{if } v_i \text{ belongs to class } c_j \\ 0 & \text{else} \end{cases} \tag{2}$$

In the label propagation process, each entity v_i will receive the label information from its neighborhoods, and retain its initial class state at time $t + 1$:

$$\boldsymbol{f}_i^{t+1} = \alpha \sum_{v_j \in N(v_i)} w_{ij} \boldsymbol{f}_j^t + (1-\alpha) \boldsymbol{f}_i^0 \tag{3}$$

where $0 < \alpha < 1$ is the influence weight of neighbors for entity v_i. Finally, we can estimate the probability of each class for each unknown entity when the above iterative procedure converges. The convergence of label propagation can be found in [3].

3   Experiments

This paper aims at extracting fine-grained entities. In line with this, we select two datasets the Wiki-list and the Web-list for comparative studies. The Wiki-list contains 30,525 entities and 73,074 types extracted from Wikipedia pages. The Web-list contains 8,724 entities and 20,593 types extracted from traditional web pages. In the gold set, there are about 3.7 types for each entity. Fig. 2 (a) and (b) show the effect of varying unlabeled ratio from 10% to 90%, with a step up size of 10%.   Clearly, the performance decreases gradually, because it is more difficult to predict with less known entities. According to Fig. 2 (a) and (b), for the Wiki-list dataset, the PMI measure can get better performance than co-occurrence measure. However, for the Web-list dataset, the PMI measure cannot achieve better performance than co-occurrence measure. In addition, the performance for the Wiki-list dataset is significant better than the performance for the Web-list dataset. We believe that it is because traditional web pages are much noisier than Wikipedia pages.
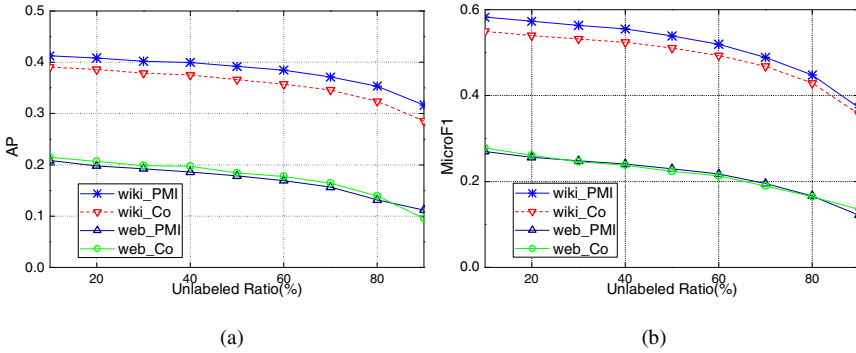
Fig. 2. Performance sensitivity to unlabeled ratio using different similarity measures

4   Related Work

Recently, there has been some work on building tagging systems using a large number of fine-grained types. Some focus on person categories (e.g. [4]). Some deal with around 100 types at most (e.g. [5]). We label entities with a great large number of types. Limaye et al. [6] annotate table cells and columns with corresponding categories and relations from an existing catalog or type hierarchy for a single table. Different from them, we assign class labels for a list corpus instead. Talukdar et al. [7] acquires labeled classes and their instances from both of unstructured and structured text sources using graph random walks. They construct the graph model based on the relationship of entities and classes. We use structured text sources and construct the graph model based on the relationship of entities co-occurrence in coordinate ways.

5   Conclusions and Future Work

We have proposed a semi-supervised label propagation fine-grained entity extraction method from an open domain. Results on two real-world datasets show that the method achieves the promising results. In future, we plan to evaluate this method with YAGO ontology which is stricter in category hierarchy than the categories in Wikipedia.

References

1. Guo, J., et al.: Named entity recognition in query. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, USA, pp. 267–274. ACM (2009)

2. Jiang, P., et al.: Wiki3C: exploiting wikipedia for context-aware concept categorization. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, Rome, Italy, pp. 345–354. ACM (2013)
3. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. In: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, Pennsylvania, pp. 985–992. ACM (2006)
4. Ekbal, A., et al.: Assessing the challenge of fine-grained named entity recognition and classification. In: Proceedings of the 2010 Named Entities Workshop, Uppsala, Sweden, pp. 93–101. Association for Computational Linguistics (2010)
5. Ling, X., Weld, D.S.: Fine-Grained Entity Recognition. In: Proceedings of the 26th Conference on Artificial Intelligence, AAAI (2012)
6. Limaye, G., Sarawagi, S., Chakrabarti, S.: Annotating and searching web tables using entities, types and relationships. Proc. VLDB Endow. 3(1-2), 1338–1347 (2010)
7. Weischedel, R., Brunstein, A.: Bbn pronoun coreference and entity type corpus. Linguistic Data Consortium, Philadelphia (2005)

NLP-Driven Event Semantic Ontology Modeling for Story

Chun-Ming Gao, Qiu-Mei Xie, and Xiao-Lan Wang

School of Information Science and Engineering, Hunan University, Changsha 410082, China
{gcm211,xieqiumei0814}@163.com,397543934@qq.com

Abstract. This paper presents a NLP-driven semantic ontology modeling for unstructured data of Chinese children stories. We use a weakly-supervised approach to capture n-ary facts based on the output of dependency parser and regular expressions. After n-ary facts post-processing, we populate the extracted facts of events to SOSDL (Story-Oriented Semantic Language), an event ontology designed for modeling semantic elements and relations of events, to form a machine-readable format. Experiments indicate the reasonability and feasibility of our approach.

Keywords: Information Extraction, Natural Language Processing, N-ary Relation, Event Ontology.

1 Introduction

Cognition psychologists consider events as the basic unit of the real world that human memory can understand. This paper defines event e as: (p, a1, a2, … , an), where p is the predicate that triggers the presence of e in text and it can't be null, while a1, a2, … , an are the arguments associated with e. Wei Wang [1] obtained 5W1H events in topic sentences from Chinese news, but it focused on the interesting information not the whole text; Yao-Hua Liu [2] derived the binary relations based on verbs from news text, but it suffered quality loss for extracting higher order n-ary facts. We use Open Information Extraction (OIE) [3] to capture n-ary fact-frames from Chinese children stories and convert them into event structures. For the complex sentences, we represent the extracted n-ary facts from relative clause as the nested structure of main fact-frame structure. The fact-frame structure is made of facts which are composed of attribute-value pairs where attribute is obtained from dependency relation from parser or regular expressions, while value is the word from sentence or entity annotated type. The set of attributes are *{subject, predicate, object, instrument, place, time, subject property, object property, subject amount, object amount, adverb}*.

Ontology is an effective knowledge organization model with the semantic expression and reasoning ability. SOSDL ontology is designed to be a common event model to describe the event information in multimedia data (text, audio, image, etc.), with the representation ability of quantitative temporal (Allen time relations [4]) and spatial relations (topologic relations, directional relations and distance relations). The top concepts are *Entity, Object, Time, Space, Information Entity, Event* and *Quality*. For

an event e, we incorporate its predicate as the individual of *Event*, its arguments as the individuals of the relevant SOSDL Classes by its fact-attributes. The relations between predicate and arguments are attached by reification technique[1], while the relations between events are ordered by event start time and duration. Then we can use web semantic technology to query and reason for the applications, like Question Answering Engines and Text-Driven Animation Generation, etc. This paper mainly focuses on the event semantic elements extraction which is the key step.

2 Event Semantic Elements Extraction

2.1 N-Ary Facts Extraction

For original text, we do word segmentation and POS tagging by ICTCALS[2], apply rules to identify named entities (characters, time, location) and Chinese verb change forms (e.g.看一看(look at),看了又看(look at for times)), treat the character dialogues as a whole part, split text into sentences, and finally save it as XML format.

Stanford Chinese Dependency Parser [5] gives 45 kinds of named grammar relations and a default relation *dep*. The output of it consists of sentences represented as binary dependencies from the head lemma to the dependent lemma: *rel (head,dep)*. An extraction example is shown in Fig.1. We have totally got 33 fact frame elements extraction rules for Chinese sentences which explicitly contain a verb or adjective as the candidate predicate, and these extraction rules are faced with main clause, relative clause, Chinese specific structures "把 (BA)", "被 (BEI)", etc.
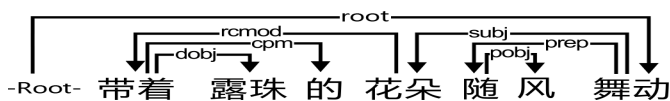
Fig. 1. Extraction example of "带着露珠的花朵随风舞动。(Flowers carrying dewdrops are dancing with the wind.)". It contains main clause and relative clause, in the following mentioned rules, ↑means the head of rel, ↓means the dep of rel. The main verb"舞动 (are dancing)" is found by *root-↓*; for *nsubj-↑*and *prep-↑*both are equal with *root-↓*, two arguments "花朵 (flowers)" and "风 (the wind)" are found by *nsubj-↓*and *prep-↓-pobj-↑*. As the relative clause, *rcomd-↓* equals *dobj-↑*, we find the subject, predicate and object by using *rcomd-↑*, *rcomd-↓* and *dobj-↓*, finally we save it as the nested frame structure of subject property of main frame.

However, Chinese sentences are composed of topics and comments, which can not contain candidate predicates but just convey a clear meaning (e.g., "他今年8岁。(He is 8 years old this year.)"). As most language theories focus on verbs, the dependency parser will fail to parse these sentences. In this case "岁 (year)" will be treated as verb not a quantifier. As a complement, we use regular expressions to extract entities

---

1 Defining N-ary Relations on the Semantic Web. http://www.w3.org/TR/swbp-n-aryRelations/

2 Chinese lexical Analysis System.http://www.ictcals.org/index.html

by annotation in those clauses without candidate predicates. Additionally, from the output of Stanford dependency parser, it is difficult to conclude the extraction rules of sentence pattern like "NP1+VP1+NP2+VP2 (+NP3)" where NP2 is the object of VP1 and the subject of VP2 (e.g., 小猴看见一只小刺猬在散步。(A Little monkey saw a little hedgehog walking.)), which do not conflict with the existing rules. Then the expression"(.*)/n(.*?)/v(.*?)/n(.*?)/v((.*?)/n)?" is used to match this sentence pattern, and two frame structures are founded by method *group(int i) where* $i \geq 1$ of Matcher object in java.util.regex: [predicate:看见(saw), subject:小猴(A Little monkey), object: 小刺猬(a little hedgehog), object amount: 一只(a)] and [predicate:散步(walking), subject:小刺猬(a little hedgehog)].

2.2   N-Ary Fact-Frame Post-processing

Compared to the even form, there are three types of n-ary fact-frame need to be modified: (1) fact-frame only contains time or location fact; (2) fact-frame has at most one subject and multi-predicates; (3) fact-frame has multi-subjects, and each subject is followed by multi-predicates. We use the following heuristics to process them as candidate event form: for case (1), as our text understanding based on event model, we attach it to next fact-frame who has predicate fact; for case (2), common senses tell us these multi-predicates are likely share a same subject, we create the candidate events as the number of predicates; for case (3), we spilt the fact-frame into sub-frames by subjects, for each sub-frame we do it as case (2). As Lexical Grammar Model [6] specifies verbs in any language can be classified into ten general domains, and each lexical domain is characterized in terms of the trigger words of a general verb or genus, we define the event types as the verb classifications and use trigger-event-type table to identify event types. Finally we populate the event elements to SOSDL.

3   Evaluation

We use 154 Chinese children stories as the data set and set up four experiments. Baseline1 is the pretreatment without identification of Chinese verb change forms. Baseline2 only contains main clauses extraction rules. Liuyaohua is the method in [2]. We apply P, R, F and completeness C to evaluate n-ary facts extraction. Let N_t be the number of facts that should be extracted from texts, N_f be the number of facts found by methods. For extracted facts, we manually judge the following numbers: 1) true and complete $N_{t\&c}$, 2) true and incomplete $N_{t\&inc}$, or 3) false. True and incomplete facts either lack arguments that are present in the sentence, or contain underspecified arguments, but are nevertheless valid statements. Let $R = (N_{t\&c} + N_{t\&inc})/N_t$,

$$R = (N_{t\&c} + N_{t\&inc})/N_f, \quad C = N_{t\&c}/N_f, \text{ and } F = 2RP(R+P).$$

Table 1. Evaluation results of System, Baseline1, Baseline2 and Liuyaohua with R, P, F and C

Method	N_f	$N_{t\&c}$	$N_{t\&inc}$	N_t	R	P	F	C
System	4078	3012	340	4344	77.16%	82.20%	79.60%	73.86%
Baseline1	4114	2700	356	4344	70.35%	74.28%	72.26%	65.63%
Baseline2	3922	2484	538	4344	69.57%	77.05%	73.12%	63.33%
Liuyaohua	3388	2149	431	4344	59.39%	76.15%	66.73%	63.42%

From Table1 we observe a significantly higher numbers of true and complete facts for system, as well as a higher overall R, P, F, and C. The R in Liuyaohua is 59.39%, and it suffers quality loss for n-ary facts extraction. It is seen that Baseline1 find out more facts than System, that's because Chinese verb change forms result to the redundant information which makes Stanford Dependency Parser lacks training for those types of sentences. From the results of system and Baseline2, we see rich extraction rules can effectively improve the completeness and recall.

4   Conclusion

We described an almost unsupervised approach for event semantic understanding task of Chinese children texts. One major drawback of our system to extract facts is that the dependency parse does not contain the dependency *dep*, which indicates unclear grammatical relationships. Additionally, wrong segmentation and POS tag may produce accumulative errors with the dependency parse, for example a noun is wrong tagged as a verb, or segmented into a verb or a noun. Future work will focus on the using very fast dependency parsers and concluding rich linguistic grammars of Chinese special structure to improve the extraction results.

References

1. Wang, W.: Chinese News Event 5WH Semantic Elements Extraction for Event Ontology Population. In: Proceedings of the 21st International Conference Companion on World Wide Web, Lyon, France, pp. 197–202 (2012)
2. Liu, Y.H.: Chinese Event Extraction Based on Syntactic Analysis. MA Thesis. Shang Hai University, China (2009)
3. Gamallo, P., Garcia, M.: Dependency-Based Open Information Extraction. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, pp. 10–18 (2012)
4. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. Communications of the ACM 26, 832–843 (1983)
5. Chang, P.C., Tseng, H., Jurafsky, D., Manning, C.D.: Discriminative Reordering with Chinese Grammatical Relations Features (2010),
   http://nlp.stanford.edu/pubs/ssst09-chang.pdf
6. Ruiz de Mendoza Ibáñez, F.J., Mairal Usón, R.: Levels of description and constraining factors in meaning construction: an introduction to the Lexical Constructional Model (2008),
   http://www.lexicom.es/drupal/files/RM_Mairal_2008_Folia_Lingu
   istica.pdf

The Development of an Ontology for Reminiscence

Collette Curry, James O'Shea, Keeley Crockett, and Laura Brown

Manchester Metropolitan University, John Dalton Building, Manchester
{collette.curry,j.d.oshea,k.crockett,laura.brown}@mmu.ac.uk

Abstract. The research presented in this paper investigates the construction and feasibility of use of an ontology of reminiscence in a conversational agent (CA) with suitable reminiscence mechanisms for non-clinical use within a healthy aging population who may have memory loss as part of normal aging and thereby improve subjective well-being (SWB).

Keywords: ontology, reminiscence, conversational agent, gerontology.

1 Introduction

1.1 Why an Ontology Is Important

The use of ontologies in computer science has been steadily emerging into the discipline over several decades. The evolution of the semantic web has encouraged the development of ontologies. This is because an ontology represents the shared understanding and the well-defined meaning of a domain of interest, thereby enabling computers and people to collaborate better [1].

1.2 Importance of Reminiscence

Reminiscence concerns telling stories of the past, personal histories, individual perceptions of social worlds inhabited, and events experienced personally or at a distance [8]. It can be pleasurable, cathartic, or therapeutic. Since publication of the Butler [9] paper there has been an exponential growth in literature concerning reminiscence and life review, making the importance of reminiscence and life review in the caring services clear.

2 Ontology: Production

2.1 Methods

An iterative approach to development of the ontology was adopted early on in the process, starting with an a priori list of instances, then revising and refining the evolving ontology and filling in the details. Once the initial version of the ontology was defined, we evaluated and debugged it by using it in the content of the conversational agent. As a result of user interaction, it was then possible to revise the initial ontology. This process of iterative design will continue through the entire lifecycle of the ontology.

1	CLASSES	Properties	
2	Adulthood	hasStatus	isAnAdult
3	Americans	overHere	isForeign
4	BackToBackHouses	twoUpTwoDown	isCold
5	BedBugs	hasInfestation	isBitten
6	BelleVueZoo	hasWildAnimals	inCaptivity
7	BetamaxTapes	recordTV	isRedundant
8	BoxBrownieCamera	isAffordable	hasFilm
9	Boyfriends	hasDateEvent	isFun
10	Brylcreem	onTheHair	calledBrylcreemBoys
11	Buses	hasBusStop	hasBusStation
12	Cameras	hasFilm	hasPhotograph
13	Childhood	hasToGrowUp	isAnInfant
14	ChildLabour	hasWorkToDo	noSchool
15	Chippy	hasChipsAndFish	hasFish
16	Christmas	hasPresents	isReligious
17	Church	hasWeatherVane	isReligious
18	Cinemas	hasMovies	isEntertainment
19	Circus	hasElephants	hasClowns
20	Clothing	latestFashion	isAMiniSkirt

Fig. 1. Classes and properties of ontology of reminiscence

3   Ontology of Reminiscence

Initially, text was written down in natural language that described the reminiscence domain. This allowed the creation of a glossary of natural language terms and definitions [9] [Figure 1]. Ontology was initially proposed by the artificial intelligence community to model declarative knowledge for knowledge-based systems, to be shared with other systems. Once the concept of ontology was defined, production of the Ontology of Reminiscence was begun. A preliminary ontology was created and mapped to the WordNet hierarchy [3:4:10:11:12], and then implemented within the

```
 1
 2   table: ~memory (^role ^memory ^kind ^item)
 3   ^createfact(^memory member ^role)
 4   if (^kind != *) {^createfact(^item member ^kind)}
 5   if (^item != *) {^createfact(^item write ^memory)}
 6   ^addproperty(^memory NOUN_MEMORY )
 7   DATA:
 8   ~news Neil_Armstrong ~event [Lunar_Landing_1969]
 9   ~news "Buzz Aldrin" * *
10   ~news "Michael Collins"  ~event "First moon landing"
```

Fig. 2. Mapping of the moon landing data

conversational agent 'Betty' (CA).   The mapping to the WordNet hierarchy was achieved by breaking down the ontology into nouns, adjectives, opposites, prepositions, verbs and concepts.   These were then mapped to elements within WordNet and scripted in the program of the CA.

The CA, 'Betty', has both short and long term memory. This means that 'Betty' can listen, talk and remember, all by using saved variables. What the user has already said during the conversation can be checked, using conditions to verify whether a variable was set, as well as the rules controlling input parameters. After each input, the CA first tried to understand the user input, then it updated the current state and generated output to the user. All inputs and outputs are appended to the log. These logs can be studied to enable the CA to be updated as required.

3.1   Experiments

This research conducted a pilot evaluation via a comparative usability test with 5 people, to explore if the CA 'Betty' effectively contributed to reminiscence in terms of its functionality and interface.   For our test, a group of five over 45-year-olds spoke with 'Betty' for a five minute period. This was to test the precision, recall, and accuracy of the CA [5]. Further experiments were run to test for user subjective well-being and memory recall improvement.   These were carried out with 30 participants aged 45+ and showed that well-being was improved by the use of the CA and that the participants recall of past events was increased.   Well being was measured before and after application of the CA by use of a general anxiety and depression scale.   The application of an Everday Memory Questionnaire (EMQ) [7] demonstrated a noticeable difference in cognitive ability after use of the CA.   This more direct assessment of the errors experienced by older adults during their daily activities may be more useful for directing the research into developing an intervention that will have a practical and therapeutic impact [13].

3.2   Conclusions and Further Work

The main motivation behind ontologies is that they allow for sharing and reuse of knowledge bodies in computational form [6]. Ontology design is a creative process, and no two ontologies designed by different people can be the same. The potential applications of the ontology and the designer's understanding and view of the domain will undoubtedly affect ontology design choices. In terms of 'Betty', assessment of the quality of the ontology by using it in the CA application for which it was designed is necessary [2]. Further work to develop an automatic method of updating the CA via the user logs, and therefore learning from the conversation, will be carried out to improve the responses from 'Betty'.

References

1. Gómez-Pérez, A.: Knowledge sharing and reuse. In: Liebowitz (ed.) Handbook of Applied Expert Systems. CRC Press, Boca Raton (1998)
2. Gruninger, M., Fox, M.S.: Methodology forthe Design and Evaluation of Ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI 1995, Montreal (1995)
3. Hendler, J., McGuinness, D.L.: The DARPA Agent Markup Language. IEEE Intelligent Systems 16(6), 67–73 (2000)
4. Humphreys, B.L., Lindberg, D.A.B.: The UML Sproject: making the conceptual connection between users and the information they need. Bulletin of the Medical Library Association 81(2), 170 (1993)
5. Nielsen, J., Landauer, T.K.: A mathematical model of the finding of usability problems. In: Proceedings of ACM INTERCHI 1993 Conference, Amsterdam, The Netherlands, April 24-29, pp. 206–213 (1993)
6. Webster, J.D.: The reminiscence functions scale: a replication. International Journal of Aging and Human Development 44(2), 137–148 (1997)
7. Wagner, N., Hassanein, K., Head, M.: Computer use by older adults. A multi-disciplinary review. Computers in Human Behaviour 26, 870–882 (2010)
8. Parker, J.: Positive communication with people who have dementia. In: Adams, T., Manthorpe, J. (eds.) Dementia Care, pp. 148–163. Arnold, London (2003)
9. Butler, R.N.: The Life Review: An interpretation of reminiscence in the aged. Psychiatry 26, 65–76 (1963)
10. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM 38(11), 39–41 (1995)
11. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
12. WordNet: An Electronic Lexical Database (citations above) is available from MIT Press, http://mitpress.mit.edu (accessed on January 14, 2013)
13. Duong, C., Maeder, A., Chang, E.: ICT-based visual interventions addressing social isolation for the aged. Studies Health Technology Inform. 168, 51–56 (2011)

Chinese Sentence Analysis Based on Linguistic Entity-Relationship Model

Dechun Yin[*]

School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
`yindechun163@163.com`

Abstract. We propose a new model called linguistic entity relationship model (LERM) for the Chinese syntactic parsing. In this model, we implement the analysis algorithm based on the analysis and verification of the linguistic entity relationship modes that are extracted and defined to describe the most basic syntactic and semantic structures. Compared with the corpus-based and rule-based methods, we neither manually write a large number of rules as used in traditional rule-based methods nor use the corpus to train the model. We only use the few meta-rules to describe the grammars. A Chinese syntactic parsing system based on the model is developed, and its performance of syntactic parsing outperforms the corpus-based baseline system.

Keywords: linguistic entity-relationship, Chinese analysis, syntactic analysis.

1 Introduction

Many rule-based and corpus-based methods have been proposed for the Chinese syntactic parsing. The rule-based methods need a large number of generation rules that are often manually edited by developers, and the corpus-based methods need a large-scale corpus to train the linguistic model. For avoiding the laborious, costly and time-consuming work of editing the rules and building the corpus, we propose the linguistic entity relationship model(LERM). In the model, we only use the few meta-rules to describe the grammars and parse the sentence.

2 Linguistic Entity Relationship Model

The linguistic entity relationship model(LERM) is used for describing the interaction of linguistic entities. In this model, we generalize and build the basic relationship modes by objectively summarizing and abstracting the Chinese grammars, and we use the modes as the foundation of the Chinese syntactic parsing.

The linguistic entity, which can be a word, phrase or sentence, is defined as the structural unit of being used for analyzing the relationship modes of linguistic entities, and it is a structural constituent of the sentence. The basic relationship modes of linguistic entities, which are used for the algorithm of analysis and verification of relationship

---

[*] Corresponding author.

modes, are extracted and generalized from the Chinese corpus with the help of Chinese grammar dictionary, and therefore they can be used to describe the most basic grammatical and semantic logic of Chinese sentence.

Generally, the relationship modes of linguistic entities are lexicalized and built on the verb, and they can describe the syntactic and semantic structure of a sentence, such as the relationship of predicate and argument. Furthermore, the relationship modes can also be built on the adjective in adjective-predicate sentence, or on the noun in noun-predicate sentence. In this paper we only present the relationship modes that are built on the verb because they are the most important and complex relationship modes compared with the others.

LERM has five most basic relationships, and each relationship includes some relationship modes. The relationship G denotes the subject-verb-object(SVO) or subject-verb(SV) sentence; D denotes the double-object sentence; C denotes the causative or imperative sentence; L includes but no limited "是" sentence; E includes but no limited "有" sentence(i.e., sentence of being or existential sentence). They are described in detail in Table 1. In the relationship modes, entity a, b, c or s is the argument. In particular, s is a special entity of being a subsentence, which presents the property of recursion of natural language. The relationship G, D, C, L or E is the predicate and is often built on the verb. However, in some special Chinese sentence, such as the adjective-predicate sentence whose relationship is G, the predicate is adjective, and the relationship modes of G only include aG and sG.

The relationship modes are lexicalized for being built on the verbs. They are semiautomatically extracted, manually edited, and stored in the linguistic entity relationship dictionary. For example, some relationship modes built on the verb "看" are described and the conceptual constraints of entities are given in Table 2.

3   Chinese Syntactic Parsing Based on LERM

We define the Chinese sentence CS as $CS = W_1W_2…W_i…W_N$, where CS totally contains N Chinese words and W_i is the ith Chinese word, and define the POS tag of the word W_i as WP_i. If the POS tag WP_i is verb, we define the meaning set of WP_i that contains m meanings as $WPM=\{WPM_1,…,WPM_i,…,WPM_m\}$ and define the relationship mode set of the meaning WPM_i that contains r relationship modes as $WPMR=\{WPMR_1,…,WPMR_i,…,WPMR_r\}$. The above knowledge of POS tags, meanings and relationship modes is stored in the linguistic entity relationship dictionary, and will be used as the input parameters of the procedure of complete analysis and verification of relationship modes.

The algorithm of parsing is described in Figure 1. In the algorithm, we analyze and verify whether each input sentence is any one of the five Chinese sentence patterns that are listed in Figure 1. For example, in the procedure of *VerbalPredicateSentence()*, the relationship modes are built on the verb, so each relationship mode of the verb of the input sentence is analyzed and verified. After being verified, the mode that is approved to be reasonable and correct can be used to describe the syntactic and semantic structure of the sentence. Each relationship mode has the corresponding analysis and verification action set(**RMAVActionSet**), which is listed in Table 1. The action expression *F(X)* and *F(Y)* in **RMAVActionSet** are recursively defined as the

few meta-rules, which are similar to the generation rules of context-free grammar(CFG). The few meta-rules are used for recursively parsing the subsequence *X* and *Y* of *CS*.

Table 1. Analysis and Verification Action Table(*AVActionTable*)

Relationship	Relationship Mode	RMAVActionSet:{F(X),F(Y)}
G	G	$\{\varnothing,\varnothing\}$
	aG	$\{E(X),E_R(Y)\}$
	sG	$\{RS(X),E_R(Y)\}$
	Gb	$\{E(X),E_R(Y)\}$
	Gs	$\{E(X),RS_R(Y)\}$
	aGb	$\{E(X),E_R(Y)\}$
	aGs	$\{E(X),RS_R(Y)\}$
	sGb	$\{RS(X),E_R(Y)\}$
	s_1Gs_2	$\{RS(X),RS_R(Y)\}$
D	$aDbc$	$\{E(X),E_R(Y)\}$
	$aDbs$	$\{E(X),E_R(Y)\}$
	$sDbc$	$\{RS(X),E_R(Y)\}$
	s_1Dbs_2	$\{RS(X),RS_R(Y)\}$
C	$aCs=aC(b*)$	$\{E(X),RS_R(Y)\}$
L	aLb	$\{E(X),E_R(Y)\}$
	aLs	$\{E(X),RS_R(Y)\}$
	sLb	$\{RS(X),E_R(Y)\}$
	s_1Ls_2	$\{RS(X),RS_R(Y)\}$
E	aEb	$\{E(X),E_R(Y)\}$
	aEs	$\{E(X),RS_R(Y)\}$
	sEb	$\{RS(X),E_R(Y)\}$

Table 2. Some Meanings and Relationship Modes of Verb "看"

Verb	Meaning	Relationship Mode	Abbreviation
看	see; look at; watch	*[**a**:(person,animate)] **G** [**b**:(person, animate,substance,abstract,location,activity)]*	*aGb*
		*[**a**:(person, animate)] **G** s*	*aGs*

4   Experiment and Evaluation

Since the corpus-based system MaltParser [1] recently shows almost state-of-the-art performance on multilingual dependency parsing tasks in comparison to the other approaches, we use it as the baseline system of Chinese parsing. We select the top 1000 sentences from Penn Chinese Treebank 5.1 to build the test dataset and use the remaining sentences as the training dataset for the MaltParser. We employ the method [2] to convert the phrase structure of the sentence into dependency structure. The experimental results are listed in Table 3.

Table 3 shows that the system LERM gets the better performance. Especially the labeled and root accuracy are encouraging. Since the entry of the parsing is the verb that is often the root node of most sentences, the parsing can take the global syntactic and semantic features into account. This ensures that the verb and its arguments are adequately analyzed and verified. As a result, the root accuracy of system LERM is remarkably higher than the baseline system, and this also benefit the improvement of the labeled accuracy.

```
Initial state of input sentence: CS=<W₁…Wᵢ₋₁WᵢWᵢ₊₁…Wₙ>(1≤i≤N)
function WholeAnalysis(CS) {
step1:VerbalPredicateSentence(CS);//verify whether CS is verbal-predicate pattern
step2:AdjPredicateSentence(CS);//verify whether CS is adjective-predicate pattern
step3:NounPredicateSentence(CS);//verify whether CS is noun-predicate pattern
step4:ConsecutiveVerbSentence(CS);// verify whether CS is consecutive-verb pattern
step5:CompositeSentence(CS); //verify whether CS is composite sentence pattern
}, Due to the limit of space, we present here only the verification of verbal-predicate sentence:
function VerbalPredicateSentence(CS){
foreach(Wᵢ of CS)          // Wᵢ is a word of CS
  if(WPᵢ of Wᵢ is verb)      // if WPᵢ is the verb
     foreach(WPMᵢ of WPᵢ)       // WPMᵢ is a meaning of verb WPᵢ
       foreach(WPMRᵢ of WPMᵢ) // WPMRᵢ is a relationship mode of meaning WPMᵢ
       {      //analysis and verification of the relationship mode WPMRᵢ
          for  current state of CS: CS=<W₁…Wᵢ₋₁>Wᵢ<Wᵢ₊₁…Wₙ> =
<X>Wᵢ<Y>
          get RMAVActionSet of relationship mode WPMRᵢ from
          AVActionTable,and then execute the action F(X) and F(Y) in
          RMAVActionSet to recursively parse the subsequence X and Y;
       }
}
```

Fig. 1. Algorithm of Parsing

Table 3. Syntax Test.(**Labeled** precision is percentage of non-root words assigned correctly heads and dependency labels; **root** precision is percentage of correctly identified root words; **complete** precision is percentage of completely matched sentence. Punctuations are excluded.)

System	Labeled(%)	Root(%)	Complete(%)
MaltParser	82.97	70.63	30.56
LERM	91.72	88.73	50.45

References

1. Nivre, J.: Algorithms for Deterministic Incremental Dependency Parsing. Computational Linguistics 34(4), 513–553 (2008)
2. Hall, J.: MaltParser-An Architecture for Labeled Inductive Dependency Parsing. Licentitate thesis, Vaxjo University, pp. 52–53 (2006)

A Dependency Graph Isomorphism
for News Sentence Searching

Kim Schouten and Flavius Frasincar

Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, The Netherlands
{schouten,frasincar}@ese.eur.nl

Abstract. Given that the amount of news being published is only increasing, an effective search tool is invaluable to many Web-based companies. With word-based approaches ignoring much of the information in texts, we propose Destiny, a linguistic approach that leverages the syntactic information in sentences by representing sentences as graphs with disambiguated words as nodes and grammatical relations as edges. Destiny performs approximate sub-graph isomorphism on the query graph and the news sentence graphs, exploiting word synonymy as well as hypernymy. Employing a custom corpus of user-rated queries and sentences, the algorithm is evaluated using the normalized Discounted Cumulative Gain, Spearman's Rho, and Mean Average Precision and it is shown that Destiny performs significantly better than a TF-IDF baseline on the considered measures and corpus.

1 Introduction

With the Web continuously expanding, humans are required to handle increasingly larger streams of news information. While skimming and scanning can save time, it would be even better to harness the computing power of modern machines to perform the laborious tasks of reading all these texts for us. In the past, several approaches have been proposed, the most prominent being TF-IDF [6], which uses a bag-of-words approach. Despite its simplicity, it has been shown to yield good performance for fields like news personalization [1]. However, the bag-of-words approach does not use any of the more advanced linguistic features that are available in a text (e.g., part-of-speech, parse tree, etc.).

In this paper we propose a system that effectively leverages these linguistic features to arrive at a better performance when searching news. The main idea is to use the dependencies between words, which is the output of any dependency parser, to build a graph representation of a sentence. Then, each word is denoted as a node in the graph, and each edge represents a grammatical relation or dependency between two words. Now, instead of comparing a set of words, we can perform sub-graph isomorphism to determine whether the sentence or part of a sentence as entered by the user can be found in any of the sentences in the database. Additionally, we implemented the simplified Lesk algorithm [4] to

perform word sense disambiguation for each node, so that it will represent the word together with its sense.

The method we propose to compare to graphs is inspired by the backtracking algorithm of McGregor [5], but is adjusted to cope with partial matches. The latter is necessary since we do not only want to find exact matches, but also sentences that are similar to our query to some extent. As such, we aim to produce a ranking of all sentences in the database given our query sentence.

2   News Searching

To compare two graphs, we traverse both the query sentence graph and each of the news sentence graphs in the database in a synchronized manner. Given a pair of nodes that are suitable to compare, we then recursively compare each dependency and attached node, assigning points based on similarity of edges and nodes. In this algorithm, any pair of nouns and any pair of verbs is deemed a proper starting point for the algorithm. Since this results in possibly more than one similarity score for this news-query sentence combination, we only retain the highest one.

The scoring function is implemented as a recursive function, calling itself with the next nodes in both the query graph and the news item graph that need to be compared. In this way, it traverses both graphs in parallel until one or more stopping criteria have been met. The recursion will stop when there are either no more nodes or edges left to compare in either or both of the graphs, or when the nodes that are available are too dissimilar to justify comparing more nodes in that area of the graph. When the recursion stops, the value returned by the scoring function is the accrued value of all comparisons made between nodes and edges from the query graph and the news item graph.

A genetic algorithm has been employed to optimize the parameters that weigh the similarity score when comparing nodes and edges. Mainly used to weigh features, an additional parameter is used to control the recursion. If there is no edge and node connected to the current node that is able to exceed this parameter, the recursion will stop in this direction.

Computing the similarity score of edges is simply done by comparing the edge labels, which denote the type of grammatical relation (e.g., subject, object, etc.). For nodes, we compare five word characteristics: stem, lemma, literal word, basic POS category (e.g., noun, verb, adjective, etc.), and detailed POS category (plural noun, proper noun, verb in past tense, etc.). These lexico-syntactic features are complemented by a check on synonymy and hypernymy using the acquired word senses and WordNet [2]. Last, by counting all stems in the database, we adjust the node score to be higher when a rare word rather than a regular word is matched.

3   Evaluation

In this section, the performance of the Destiny algorithm is measured and compared with the TF-IDF baseline. To that end, we have created a database of 19

news items, consisting of 1019 sentences in total, and 10 query sentences. All possible combinations of query sentence and news sentence were annotated by at least three different persons and given a score between 0 (no similarity) and 3 (very similar). Queries are constructed by rewriting sentences from the set of news item sentences. In rewriting, the meaning of the original sentence was kept the same as much as possible, but both words and word order were changed (for example by introducing synonyms and swapping the subject-object order). The results are compared using the normalized Discounted Cumulative Gain (nDCG) over the first 30 results, Spearman's Rho, and Mean Average Precision (MAP). Since the latter needs to know whether a result is relevant or not, and pairs of sentences are marked with a score between 0 and 3, we need a cut-off value: above a certain similarity score, a result is deemed relevant. Since this is a rather arbitrary decision, the reported MAP is the average MAP over all possible cut-off values with a step size of 0.1, from 0 to 3.

3.1   Quality of Search Results

In order to assess our solution's performance, it is compared with a TF-IDF baseline on three measures. Each of the measures is computed using the user-rated sentence pairs as the golden standard. Table 1 shows the results of all three tests, clearly demonstrating that Destiny significantly outperforms the TF-IDF baseline. The p-value for nDCG and Spearman's Rho is computed for the paired one-sided t-test on the two sets of scores consisting of the 32 split scores for both Destiny and TF-IDF, respectively. For MAP, because we computed the average over all cut-off values, the same t-test is computed over 30 cut-off values $\times$ 32 folds which results in 960 split scores.

Table 1. Evaluation results

	TF-IDF mean score	Destiny mean score	rel. improvement	t-test p-value
nDCG	0.238	0.253	11.2%	< 0.001
MAP	0.376	0.424	12.8%	< 0.001
Sp. Rho	0.215	0.282	31.6%	< 0.001

4   Concluding Remarks

Our implementation of Destiny shows the feasibility of searching news sentences in a linguistic fashion, as opposed to using a simple bag-of-words approach. By means of a natural language processing pipeline, both news items and queries are processed into graphs, which are subsequently compared to each other, with the degree of sub-graph isomorphism as a proxy for similarity. Because this graph-representation preserves much of the original semantic relatedness between words, the search engine is able to utilize this information. Furthermore,

words are not only compared on a lexico-syntactic level, but also on a semantic level by means of the word senses as determined by the word sense disambiguation implementation. This also allows for checks on synonymy and hypernymy between words. Last, the performance results on the Mean Average Precision, Spearman's Rho, and normalized Discounted Cumulative Gain demonstrate the significant gain in search results quality when using Destiny compared to TF-IDF.

Interesting topics for future work include the addition of named entity recognition and co-reference resolution to match multiple referrals to the same entity even though they might be spelled differently. Our graph-based approach would especially be suitable for an approach to co-reference resolution like [3], as it also utilizes dependency structure to find the referred entities.

References

1. Ahn, J., Brusilovsky, P., Grady, J., He, D., Syn, S.Y.: Open User Profiles for Adaptive News Systems: Help or Harm? In: 16th International Conference on World Wide Web (WWW 2007), pp. 11–20. ACM (2007)
2. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press (1998)
3. Haghighi, A., Klein, D.: Coreference Resolution in a Modular, Entity-Centered Model. In: Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2010), pp. 385–393. ACL (2010)
4. Kilgarriff, A., Rosenzweig, J.: English senseval: Report and results. In: 2nd International Conference on Language Resources and Evaluation (LREC 2000), pp. 1239–1244. ELRA (2000)
5. McGregor, J.J.: Backtrack Search Algorithms and the Maximal Common Subgraph Problem. Software Practice and Experience 12(1), 23–34 (1982)
6. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill (1983)

Unsupervised Gazette Creation
Using Information Distance

Sangameshwar Patil, Sachin Pawar, Girish K. Palshikar, Savita Bhat,
and Rajiv Srivastava

Tata Research Development and Design Centre, TCS
54B, Hadapsar Industrial Estate, Pune 411013, India

Abstract. *Named Entity extraction (NEX)* problem consists of auto-
matically constructing a gazette containing instances for each NE of in-
terest. NEX is important for domains which lack a corpus with tagged
NEs. In this paper, we propose a new unsupervised (bootstrapping) NEX
technique, based on a new variant of the Multiword Expression Distance
(MED) [1] and information distance [2]. Efficacy of our method is shown
using comparison with BASILISK and PMI in agriculture domain. Our
method discovered 8 new diseases which are not found in Wikipedia.

Keywords: Named entity extraction, Information extraction, Unsuper-
vised learning, Information distance, Agriculture.

1   Introduction

The problem of information extraction for agriculture is particularly important [1]
as well as challenging due to non-availability of any tagged corpus. Several
domain-specific *named entities (NE)* occur in the documents (such as news)
related to the agriculture domain: CROP (names of the crop including vari-
eties), DISEASE (names of the crop diseases and disease causing agents such as
bacteria, viruses, fungi, insects etc.) and CHEMICAL_TREATMENT (names of
pesticides, insecticides, fungicides etc.). *NE extraction (NEX)* problem consists
of automatically constructing a gazette containing example instances for each
NE of interest. In this paper, we propose a new bootstrapping approach to NEX
and demonstrate its use for creating gazettes of NE in the agriculture domain.
Apart from the new application domain (agriculture) for NE extraction, most
important contribution of this paper is : use of a new variant of the information
distance [2], [1] to decide whether a candidate phrase is a valid instance of the
NE or not.

2   Information Distance for NE

 [1] presented a variant of the information distance, which measures the distance
between an *n*-gram and its semantics. In this paper, we use a variant of MED to

---

[1] For instance, in India, agriculture contributes 15% of GDP and 52% of rural em-
ployment. (Source: `en.wikipedia.org/wiki/Economy_of_India`).

perform NEX. Let $\mathbf{D}$ be a given untagged corpus of sentences. Let K be a given constant indicating the window size (e.g., $K = 3$). Let g be a given candidate phrase. The *context* of g and a given word w, denoted $\phi_K(g, w)$, is the set of all sentences in $\mathbf{D}$ which contain both g (as an n-gram) and w and w occurs within a window of size K around g in that sentence. The *semantics* of g and a given word w, denoted $\mu(g, w)$, is the set of all sentences in $\mathbf{D}$ which contain both g (as an n-gram) and w, though g and w need not be within a window of size K in the sentence. Clearly, $\phi_K(g, w) \subseteq \mu(g, w)$. Then we define the *distance* between g and a given word w as follows:

$$MED0_{D,K}(g, w) = log|\mu(g, w)| - log|\phi_K(g, w)|$$

Let $W = \{w_1, w_2, \ldots, w_m\}$ be a given finite, non-empty set of m words. The definition of $MED0$ is extended to use a given set W by taking the average of the $MED0$ distance between g and each word in W:

$$MED_{D,K}(g, w) = \frac{MED0_{D,K}(g, w_1) + \ldots + MED0_{D,K}(g, w_m)}{m}$$

3 Unsupervised Gazette Creation Using MED

In unsupervised gazette creation, we are given (i) an untagged corpus $\mathbf{D}$ of documents; and (ii) a seed list L containing known examples of a particular NE type T. The goal is to create a gazette containing other instances of the NE type T that occur in $\mathbf{D}$.

The first task is to identify all phrases in $\mathbf{D}$ that are likely to be instances of the NE type T as a **pre-processing step**. Maximum Entropy (MaxEnt) classifier is trained using instances in L [2] and all the noun phrases in corpus are classified in two classes - of type T and not of type T. Phrases which are classified with higher confidence are again added to the initial training data and a new MaxEnt classifier is trained. This process of classifying noun phrases and updating MaxEnt classifier is repeated till desired number of phrases are classified as of type T. Next key step is to find "Backdrop of a Gazette", which is defined as a set W of words "characteristic" (or strongly indicative) of T. The idea is to use the $MED_{D,K}$ defined above to accept only those g which have "low" distance ("high" similarity) between g and the backdrop of the gazette L. Function $GetBackdrop(\mathbf{D}, L, K, m_0)$ (Fig 1(a)) computes, using $\mathbf{D}$, the set W for a given gazette L.

The **algorithm** $CreateGazetteMED$ (Fig 1(b)) starts with an initial seed list L of instances of a particular NE type T and a list C of candidate phrases produced by the pre-processing step. Then in each iteration, it calls the algorithm $GetBackdrop$ to create the set W of backdrop words for T using L. Then it uses the modified MED to measure the similarity of each candidate phrase $g \in C$ with W, adding g to a temporary set A only if it has a "high" similarity with W (above a threshold). At the end of $maxIter$, final set of candidates in L is

---

[2] Positive instances for other NE types play role of negative instances for classifier.

then pruned by the post-processing step. The **post-processing step** consists of *assessor* to assess and improve the quality of the candidate gazette created, by identifying (and removing) those entries in the candidate gazette which are very unlikely to be true instances of NE type T. We use a set of cue words for the NE type T and perform a statistical hypothesis test (called *proportion test*).
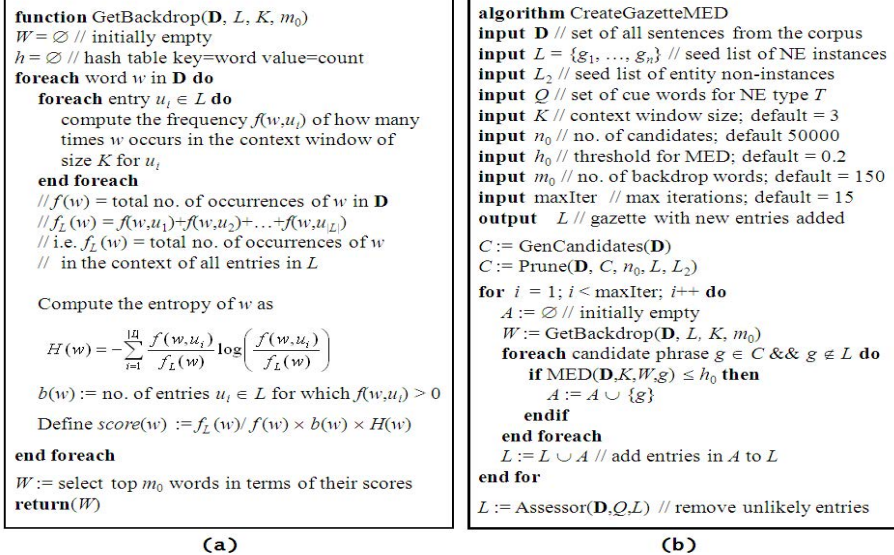
```
function GetBackdrop(D, L, K, m₀)
  W = ∅ // initially empty
  h = ∅ // hash table key=word value=count
  foreach word w in D do
    foreach entry uᵢ ∈ L do
      compute the frequency f(w,uᵢ) of how many
      times w occurs in the context window of
      size K for uᵢ
    end foreach
    // f(w) = total no. of occurrences of w in D
    // fₗ(w) = f(w,u₁)+f(w,u₂)+...+f(w,u|L|)
    // i.e. fₗ(w) = total no. of occurrences of w
    //   in the context of all entries in L

    Compute the entropy of w as
```

$$H(w) = -\sum_{i=1}^{|L|} \frac{f(w,u_i)}{f_L(w)} \log\left(\frac{f(w,u_i)}{f_L(w)}\right)$$

```
    b(w) := no. of entries uᵢ ∈ L for which f(w,uᵢ) > 0

    Define score(w) := fₗ(w)/f(w) × b(w) × H(w)

  end foreach
  W := select top m₀ words in terms of their scores
  return(W)
```

```
algorithm CreateGazetteMED
input  D // set of all sentences from the corpus
input  L = {g₁, ..., gₙ} // seed list of NE instances
input  L₂ // seed list of entity non-instances
input  Q // set of cue words for NE type T
input  K // context window size; default = 3
input  n₀ // no. of candidates; default 50000
input  h₀ // threshold for MED; default = 0.2
input  m₀ // no. of backdrop words; default = 150
input  maxIter  // max iterations; default = 15
output  L // gazette with new entries added

C := GenCandidates(D)
C := Prune(D, C, n₀, L, L₂)
for  i = 1; i < maxIter; i++ do
  A := ∅ // initially empty
  W := GetBackdrop(D, L, K, m₀)
  foreach candidate phrase g ∈ C && g ∉ L do
    if MED(D,K,W,g) ≤ h₀ then
      A := A ∪ {g}
    endif
  end foreach
  L := L ∪ A // add entries in A to L
end for

L := Assessor(D,Q,L) // remove unlikely entries
```

(a)   **(b)**

Fig. 1. Algorithms (a) *GetBackDrop* and (b) *CreateGazetteMED*

4   Experimental Evaluation

The benchmark corpus consists of 30533 documents in English containing 999168 sentences and approximately 19 million words. It was collected using crawler4j[3] by crawling the agriculture news websites such as the FarmPress group[4]. Some of the seeds used for each NE type are as follows:

- CROP: wheat, cotton, corn, soybean, strawberry, tomato, bt cotton
- DISEASE: sheath blight, wilt, leaf spot, scab, rot, rust, nematode
- CHEMICAL_TREATMENT: di-syston, metalaxyl, keyplex, evito, verimark

Starting with the candidate list C and the initial seed list for T, the algorithm *CreateGazetteMED* iteratively created the final set of 500 candidates based on $MED_{D,K}$. The post-processing step is used to further prune this list to create the final gazette for the NE type T.

Gazette sizes for each NE type are shown in Fig 2(a). Detection rate of for each NE is shown in Fig 2(b). Assessor improves precision for all NE types

---

[3] code.google.com/p/crawler4j/ an open source web crawler by Yasser Ganjisaffar.
[4] Permission awaited from the content-owners for public release of the corpus.

for both measures $MED_{D,K}$ and PMI. We compare the proposed algorithm with BASILISK [3]. Also, to gauge the effictiveness of $MED_{D,K}$ as a proximity measure, we compare it with PMI. To highlight effectiveness of the gazettes created, we compared our DISEASE gazette with wikipedia. It was quite encouraging to find that, our gazette, though created on a limited size corpus, contained diseases/pathogens not present in Wikipedia.[5] Some of these are
- `limb rot, grape colaspis, black shank, glume blotch, mexican rice borer, hard lock, seed corn maggot, green bean syndrome.`

	Crop	Disease	Chem. Treatment
$MED_{D,K}$ with assessor	322 (0.397)	389 (0.710)	398 (0.720)
PMI with assessor	171 (0.340)	152 (0.447)	398 (0.108)
$MED_{D,K}$ without assessor	499 (0.287)	490 (0.567)	500 (0.616)
PMI without assessor	493 (0.138)	485 (0.142)	500 (0.159)
BASILISK	322 (0.283)	389 (0.699)	398 (0.298)

(a)   (b)
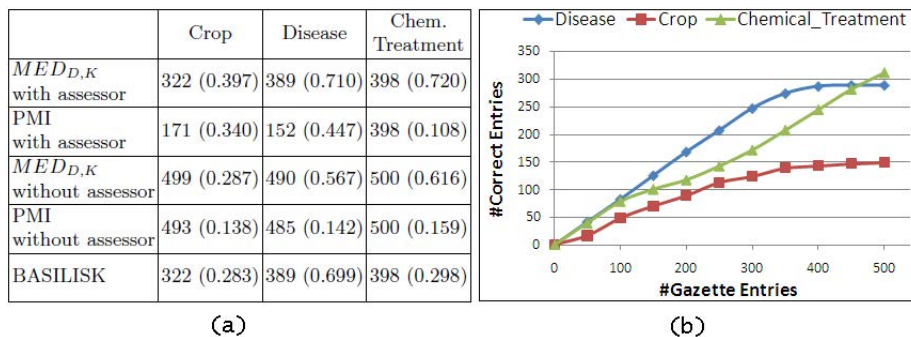
Fig. 2. (a) Number of entries in the final gazette for each NE type. (To use the same baseline for comparing precision of the proposed algorithm and BASILISK, we use the gazette size of BASILISK same as that of $MED_{D,K}$ with Assessor.) (b) Detection rate of $CreateGazetteMED$ with Assessor.

5   Conclusions

In this paper, we proposed a new unsupervised (bootstrapping) NEX technique for automatically creating gazettes of domain-specific named entities. It is based on a new variant of the Multiword Expression Distance (MED) [1]. We also compared the effectiveness of the proposed method with PMI, BASILISK [3] To the best of our knowledge, this is the first time that NEX techniques are used for the agricultural domain.

References

1. Bu, F., Zhu, X., Li, M.: Measuring the non-compositionality of multiword expressions. In: Proc. of the 23rd Conf. on Computational Linguistics, COLING (2010)
2. Bennett, C., Gacs, P., Li, M., Vitanyi, P., Zurek, W.: Information distance. IEEE Transactions on Information Theory 44(4), 1407–1423 (1998)
3. Thelen, M., Riloff, E.: A bootstrapping method for learning seman-tic lexicons using extraction pattern contexts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2002 (2002)

---

[5] Verified on 30th January, 2013.

A Multi-purpose Online Toolset for NLP Applications[*]

Maciej Ogrodniczuk and Michał Lenart

Institute of Computer Science, Polish Academy of Sciences

Abstract. This paper presents a new implementation of the multi-purpose set of NLP tools for Polish, made available online in a common web service framework. The tool set comprises a morphological analyzer, a tagger, a named entity recognizer, a dependency parser, a constituency parser and a coreference resolver. Additionally, a web application offering chaining capabilities and a common BRAT-based presentation framework is presented.

1 Introduction

The idea of making a linguistic toolset available online is not new; among other initiatives, it has been promoted by CLARIN[1], following its aspirations for gathering Web services offering language processing tools [3] or by related initiatives such as WebLicht.

The first version of a toolset for Polish made available in the Web service framework has been proposed in 2011, and called *the Multiservice* [2]. Its main purpose was to provide a consistent set of mature annotation tools — previously tested in many offline contexts, following the open-source paradigm and under active maintenance — offering basic analytical capabilities for Polish.

Since then, the Multiservice has been thoroughly restructured and new linguistic tools have been added. The framework currently features a morphological analyzer *Morfeusz PoliMorf*, two disambiguating taggers *Pantera* and *Concraft*, a shallow parser *Spejd*, the *Polish Dependency Parser*, a named entity recognizer *Nerf* and a coreference resolver *Ruler*.

2 Architecture

The Multiservice allows for chaining requests involving integrated language tools: requests to the Web service are enqueued and processed asynchronously, which allows for processing larger amounts of text. Each call returns a token used to check the status of the request and retrieve the result when processing completes.

---

[1] Common Language Resources and Technology Infrastructure, see `www.clarin.eu`

One of the major changes in the current release is a redesign of the internal architecture with the Apache Thrift framework (see `thrift.apache.org`, [1]), used for internal communication across the service. It features a unified API for data exchange and RPC, with automatically generated code for the most popular modern programming languages (including C++, Java, Python, and Haskell), the ability to create a TCP server implementing such an API in just a few lines of code, no requirement of using JNI for communication across various languages (unlike in UIMA), and much better performance than XML-based solutions.

The most important service in the infrastructure is the Request Manager, using a Web Service-like interface with the Thrift binary protocol instead of SOAP messages. It accepts new requests, saves them to the database as Thrift objects, keeps track of associated language tools, selects the appropriate ones for completing the request, and finally invokes each of them as specified in the request and saves the result to the database.

Since the Request Manager service runs as a separate process (or even a separate machine), it can potentially be distributed across multiple machines or use a different DBMS without significant changes to other components. The service can easily be extended to support communication APIs other than SOAP or Thrift and the operation does not create significant overhead (sending data using Apache Thrift binary format is much less time-consuming than sending XMLs or doing actual linguistic analysis of texts).

Requests are stored in `db4o` — an object oriented database management system which integrates smoothly with regular Java classes. Each arriving request is stored directly in the database, without any object-relational mapping code.

Language tools run as servers listening to dedicated TCP ports and may be distributed across multiple machines. There are several advantages of such architecture, the first of which is its scalability — when the system is under heavy load, it is relatively easy to run new service instances. Test versions of the services can be used in a request chain without any configuration — there is simply an optional request parameter that tells the address and port of the service. Plugging-in new language tools is equally easy — Apache Thrift makes it possible to create a TCP server implementing a given RPC API in just a few lines of code.

3   Usage and Presentation

The tools offer two interchangeable formats, supporting chaining and uniform presentation of linguistic results: TEI P5 XML and its JSON equivalent. The TEI P5 format is a packaged version of the stand-off annotation used by the National Corpus of Polish (NKJP [4]), extended with new annotation layers originally not available in NKJP.

Sample Python and Java clients for accessing the service have been implemented. To facilitate non-programming experiments with the toolset, a simple Django-based Web interface (see Fig. 1) is offered to allow users to create toolchains and enter texts to be processed.
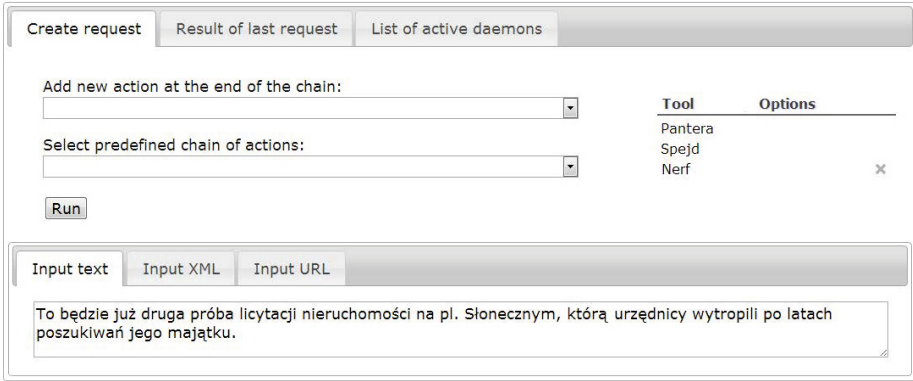
Fig. 1. The Multiservice Web interface

Fig. 2. Different levels of linguistic annotation displayed with BRAT

The application allows for triggering a processing request and periodically checking its status. Upon completion, the result is retrieved and displayed to the user. In the case of failure, an appropriate error message is presented.

The Web Interface[2] features consistent visualization of linguistic information produced with the BRAT tool [5] for all layers made available by integrated

---

[2] Available at `http://glass.ipipan.waw.pl/multiservice/`

annotators. See Fig. 2 for a selection of annotations produced for an example sentence (*Maria od zawsze kochała Jana. Gdy poprosił ją o rękę, była szczęśliwa.* 'Maria has always loved John. When he asked her to marry him, she was happy.'). Additional context-dependent linguistic properties of the annotation items (e.g. all morphosyntactic interpretations of a word, not just a disambiguated one) are available at mouseover. A unified framework for linking visualization to other levels of linguistic annotation is also provided and the only necessary implementation step is a conversion of JSON-encoded request results into BRAT internal format.

4   Conclusions

As compared to its offline installable equivalents, the toolset provides users with access to the most recent versions of tools in a platform-independent manner and without any configuration. At the same time, it offers developers a useful and extensible demonstration platform, prepared for easy integration of new tools within a common programming and linguistic infrastructure. We believe that the online toolset will find its use as a common linguistic annotation platform for Polish, similar to positions taken by suites such as Apache OpenNLP or Stanford CoreNLP for English.

References

1. Agarwal, A., Slee, M., Kwiatkowski, M.: Thrift: Scalable cross-language services implementation. Tech. rep., Facebook (April 2007),
   `http://thrift.apache.org/static/files/thrift-20070401.pdf`
2. Ogrodniczuk, M., Lenart, M.: Web Service integration platform for Polish linguistic resources. In: Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012, pp. 1164–1168. ELRA, Istanbul (2012)
3. Ogrodniczuk, M., Przepiórkowski, A.: Linguistic Processing Chains as Web Services: Initial Linguistic Considerations. In: Proceedings of the Workshop on Web Services and Processing Pipelines in HLT: Tool Evaluation, LR Production and Validation (WSPP 2010) at the 7th Language Resources and Evaluation Conference (LREC 2010), pp. 1–7. ELRA, Valletta (2010)
4. Przepiórkowski, A., Bańko, M., Górski, R.L., Lewandowska-Tomaszczyk, B. (eds.): Narodowy Korpus Języka Polskiego. Wydawnictwo Naukowe PWN, Warsaw (2012)
5. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: BRAT: a web-based tool for NLP-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2012, pp. 102–107. Association for Computational Linguistics, Stroudsburg (2012)

A Test-Bed for Text-to-Speech-Based Pedestrian Navigation Systems

Michael Minock[1], Johan Mollevik[2], Mattias Åsander[2], and Marcus Karlsson[2]

[1] School of Computer Science and Communication (CSC),
Royal Institute of Technology KTH, Stockholm, Sweden
[2] Department of Computing Science, Umeå University, Umeå, Sweden

Abstract. This paper presents an Android system to support eyes-free, hands-free navigation through a city. The system operates in two distinct modes: *manual* and *automatic*. In manual, a human operator sends text messages which are realized via TTS into the subject's earpiece. The operator sees the subject's GPS position on a map, hears the subject's speech, and sees a 1 fps movie taken from the subject's phone, worn as a necklace. In automatic mode, a programmed controller attempts to achieve the same guidance task as the human operator.

We have fully built our manual system and have verified that it can be used to successfully guide pedestrians through a city. All activities are logged in the system into a single, large database state. We are building a series of automatic controllers which require us to confront a set of research challenges, some of which we briefly discuss in this paper. We plan to demonstrate our work live at NLDB.

1 Introduction

The automated generation of route directions has been the subject of many recent academic studies (See for example the references in [1], or the very recent works [2,3]) and commercial projects (e.g. products by Garmin, TomTom, Google, Apple, etc.). The pedestrian case (as opposed to the automobile case) is particularly challenging because the location of the pedestrian is not just restricted to the road network and the pedestrian is able to quickly face different directions. In addition, the scale of the pedestrian's world is much finer, thus requiring more detailed data. Finally the task is complicated by the fact that the pedestrian, for safety, should endeavor to keep their eyes and hands free – there is no room for a fixed dashboard screen to assist in presenting route directions. We take this last constraint at full force – in our prototype there is no map display; the only mode of presentation is text-to-speech instruction heard incrementally through the pedestrian's earpiece.

We present a system to support eyes-free, hands-free navigation through a city[1]. Our system operates in two distinct modes: *manual* and *automatic*. In manual

mode, an operator guides a subject via text-to-speech commands to destinations in the city. The operator, working at a stationary desktop, receives a stream of GPS, speech and camera image data from the subject which is displayed in real time to the operator (see figure 1). In turn the operator types quick text messages to guide the subject to their destination. The subject hears the operator's instructions via the text-to-speech engine on their Android. In automatic mode the human operator is missing, replaced by a programmed controller.
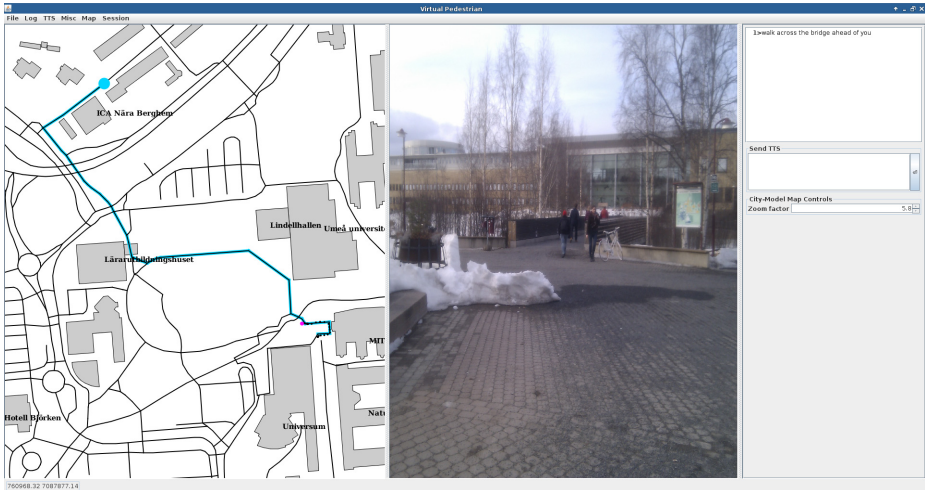
Fig. 1. Operator's interface in manual mode guiding a visitor to ICA Berghem, Umeå

The technical specification and design of our system, with an initial reactive controller, is described in a technical report [1]. That report gives a snap-shot of our system as of October 2012. In the ensuing months we have worked to optimize, re-factor and stabilize the system in preparation for its open source release – working name JANUS (Interested readers are encouraged to contact us if they wish to receive a beta-release). We have also further developed the infrastructure to integrate FreeSWITCH for speech and some extra mechanism to handle image streams. Finally we have added a facility that logs phone pictures to PostgreSQL BLOBs, the TTS messages to PostgreSQL text fields, and the audio-streams to files on the file system. Aside for server-side PL/pgSQL functions, the system is written exclusively in Java and it uses ZeroC ICE for internal communication. Detailed install instructions exists for Debian "wheezy".

2 Field Tests

We have carried out field tests since late Summer 2012. The very first tests were over an area that covered the Umeå University campus extending North to Mariahem (An area of roughly 4 square kilometers, 1788 branching points,

3027 path segments, 1827 buildings). For a period of several weeks, the first author tested the system 3-4 times per week while walking or riding his bicycle to work and back. The system was also tested numerous times walking around the Umeå University campus. A small patch of the campus immediately adjacent to the MIT-Huset was authored with explicit phrases, overriding the automatically generated phrases of a primitive NLG component (see the example in [1]). These initial tests were dedicated to validating capabilities and confirming bug fixes and getting a feel for what is and is not important in this domain. For example problems like the quantity and timing of utterances (too much or too little speech, utterances issued too late or too early) and oscillations in the calculation of facing direction led to a frustrating user experience. Much effort was directed toward fixing parameters in the underlying system, adding further communication rules and state variables, etc.

In addition to these tests, in November 2012 we conducted an initial test of our manual interface in Edinburgh (our database covered an area of roughly 5 square kilometers, 4754 branching points, 9082 path segments, 3020 buildings) – walking the exact path used in the Edinburgh evaluations of the initial SPACEBOOK prototype developed by SPACEBOOK partners Heriot-Watt and Edinburgh University [2]. With the PHONEAPP running in Edinburgh and all back-end components running in Umeå, the latencies introduced by the distance did not render the system inoperable. Note that we did not test the picture capability at that time, as it had not yet been implemented.

Due to the long Winter, we have conducted only a few outdoor tests with the system from November 2012 to April 2013. What experiments we have run, have been in an area surrounding KTH in Stockholm (An area slightly over 2 square kilometers, 1689 branching points, 3097 path segments, 542 buildings), the center of Åkersberga, and continued tests on the Umeå University campus. With the warming of the weather we look forward to a series of field tests and evaluations over the Spring and Summer of 2013.

3   System Performance

Our optimization efforts have been mostly directed at minimizing latencies and improving the performance of map rendering in our virtual pedestrian/tracking tool. There are three latencies to consider from the PHONEAPP to the controller (GPS report, speech packet, image) and one latency to consider from the controller to the PHONEAPP (text message transmission). We are still working on reliable methods to measure these latencies and, more importantly, their variability. In local installations (e.g. back-end components and PHONEAPP running in Umeå) the system latencies are either sub-second or up to 1-2 seconds – a perfectly adequate level of performance. Running remotely (e.g. back-end components running in Umeå and PHONEAPP in Edinburgh) appears to simply add a fixed constant to all four latencies.

All the map data is based on XML exports of OPENSTREETMAP data converted to SQL using the tool osm2sb (see [1]). We have limited our attention

to what may be downloaded as XML exports via OPENSTREETMAP's web-site. This has covered large enough portions of the city for our purposes. That said, we strongly believe that inefficient access to larger maps is not a significant risk.

4 Some Future Challenges

Evaluations: We have a very natural metric of evaluation: *what is a controller's effectiveness in actually guiding pedestrians from a given initial position to a given destination position?* To minimize expense, we will first employ what we term *auto evaluation*. In auto evaluation one generates random tours, unknown to the subject, over a large number of possible destinations. Because destinations are hidden, even if one of the authors serves as a subject, we will gain insight into the relative effectiveness of various controller strategies. Only after performing this cheaper form of evaluation shall we carry out a larger classical evaluations with testable hypotheses, large cohorts of random subjects, control groups, etc.

Scheduling of Utterances in Synchronization with User Position: Early in our testing we found that scheduling of utterances in synchronization with user position is a critical capability that is not easily finessed in a reactive controller. Thus we have started work on the challenging problem of predicting user position and scheduling utterances accordingly. This is briefly discussed in [4] and will be presented in greater detail in a future conference paper.

Reuse of Operator Utterances in Automatic Controllers: Our current controllers fetch pre-compiled utterances populated via primitive NLG routines run off-line. While we will explore techniques to integrate run-time NLG systems, we are interested in techniques to re-use utterances expressed by human operators (in manual mode) within our automatic controllers. We seek large collections of human authored utterance variations, where, given a large number of user trials, we might learn a policy to select when and where to issue utterances to maximize expected utility over our metric of evaluation.

References

1. Minock, M., Mollevik, J., Åsander, M.: Towards an active database platform for guiding urban pedestrians. Technical Report UMINF-12.18, Umeå University (2012),
   https://www8.cs.umu.se/research/uminf/index.cgi?year=2012&number=18
2. Janarthanam, S., Lemon, O., Liu, X., Bartie, P., Mackaness, W., Dalmas, T., Goetze, J.: A spoken dialogue interface for pedestrian city exploration: integrating navigation, visibility, and question-answering. In: Proc. of SemDial 2012, Paris, France (September 2012)
3. Boye, J., Fredriksson, M., Götze, J., Gustafson, J., Königsmann, J.: Walk this way: Spatial grounding for city exploration. In: Proc. 4th International Workshop on Spoken Dialogue Systems, IWSDS 2012, Paris, France (November 2012)
4. Minock, M., Mollevik, J.: Prediction and scheduling in navigation systems. In: Proceedings of the Geographic Human-Computer Interaction (GeoHCI) Workshop at CHI (April 2013)

Automatic Detection of Arabic Causal Relations

Jawad Sadek

School of Computing Science & Engineering, University of Salford, Manchester, England
j.sadek@hotmail.com

Abstract. The work described in this paper is about the automatic detection and extraction of causal relations that are explicitly expressed in Modern Standard Arabic (MSA) texts. In this initial study, a set of linguistic patterns was derived to indicate the presence of cause-effect information in sentences from open domain texts. The patterns were constructed based on a set of syntactic features which was acquired by analyzing a large untagged Arabic corpus so that parts of the sentence representing the *cause* and those representing the *effect* can be distinguished. To the best of researchers knowledge, no previous studies have dealt this type of relation for the Arabic language.

Keywords: Text Mining, Pattern Matching, Discourse Analysis, Information Extraction.

1   Introduction

Most studies on mining semantic relations focused on the detection of causal relations as they are fundamental in many disciplines including text generation, information extraction and question answering systems. Furthermore, they closely relate to other relations such as temporal and influence relations. These studies attempt to locate causation in texts using two main approaches; hand-coded patterns [1, 2] and machine learning approaches that aim to construct syntactic patterns automatically [3]. However, the later has exploited knowledge resources available for the language they addressed, such as large annotated corpora, WordNet and Wikipedia. Unfortunately, Arabic Language, so far, lacks mature knowledge base resources upon which machine learning algorithms can rely. In this work a set of patterns was identified based on combinations of cue words and part of speech (POS) labels which tend to appear in causal sentences. The extracted patterns reflect strong causation relations and can be very useful in the future for systems adopting machine learning techniques in acquiring patterns that indicate causation. The current study has been developed predominantly to locate intrasentential casual relations and this is believed to enhance the performance of the previous system for answering "why" question which was based on finding causality across sentences [4].

2   Arabic Causative Construction

2.1   Expression of Causation in Text

Huskour in [5] extensively surveyed causal relations in written Arabic literature; she argued that the causation from the perspective of grammarians can be classified into

two main categories. The first one is the ***verbal causality*** which can be captured by the presence of nominal clauses e.g. المفعول لأجله (Accusatives of purpose) or by causality connectors such as [لذا (therefore) – من اجل (for)] although these connectors may in many cases signal different relations other than causation. The second category is the ***context-based causality*** that can be inferred by the reader using his/her general knowledge without locating any of the previous indicators. This category includes various Arabic stylistic structures and it is frequently used in rhetorical expressions especially in novels, poetry and the Holy Quran.

2.2   Identifying Causal Relation

The definition of implicit causal relations in Arabic has been controversial among linguists and raised many interpretation and acceptance issues. It is not the aim of this paper to add to these controversies but the study will be restricted to the extraction of explicit relations indicated by ambiguous/unambiguous markers. Alternberg's typology of causal linkage was of great importance for extracting causal relation in English. Unfortunately, no such list exists for the Arabic language. Hence, a list of Arabic causal indicators needs to be created. All grammarians perspective's causative connectors mentioned in [5] have been surveyed alongside with the verbs that are synonymous with the verb "يسبب" (cause) such as " يؤدي، ينتج ... " in addition to some particles that commonly used in modern Arabic such as "حيث".

3   Constructing the Linguistic Patterns

The patterns were generated by analyzing a data collection extracted from a large untagged Arabic corpus called ***arabiCorpus[1]***. The patterns development process based on the same techniques as those used in [1]; it went through several steps of inductive and deductive reasoning methods. The two phases were assembled into single circular so that the patterns continually cycled between them until finally a set of general patterns was reached.

● **Inductive Phase:** The initial step which involves making specific observations on a sample of sentences containing causal relations retrieved from the corpus, and then detecting regularities and features in the sentences that indicate causation. This leads to formulate some tentative patterns specifying cause and effects slots. For example pattern (2) was constructed from sentence (1) specifying that the words preceding (بسبب) represent the *effect* part while the words following it represent the *cause*.

(1)   أجلت "ناسا" أمس هبوط مكوك الفضاء اتلانتس **وذلك نظرا** لسوء الأحوال الجوية.

  "NASA has postponed landing of the space shuttle Atlantis yesterday due to bad weather"

(2) R   (&C) [C] AND &This نظرا+ [E] &.

● **Deductive Phase:** involves examining the patterns that have been formulated in the previous step. At this stage the patterns are applied to the new text fragments extracted from the corpus. Three types of errors may be returned upon conducting the

---

[1] http://arabicorpus.byu.edu/search.php

patterns validation. Each kind of errors is handled by performing another inductive step:

■ *Undetected Relation*: this error occurs when the constructed patterns are unable to locate the presence of a causal relation in a text fragment. To fix this error, more patterns need to be added so that the missing relation can be identified; in some cases it may be better to modify a pattern to cover all the absent relations by omitting some of its features. For example, pattern (2) would miss the casual relation presented in sentence (3) for omitting the word "نظرا". Hence, the new pattern (4) was added.

(3)   اولت الحكومة اهتماما كبيرا لتطوير القطاع الزراعي **وذلك** رغبة منها بتحقيق الامن الغذائي
"Government paid great attention to the development of agriculture to achieve food security"

(4) R   (&C) [C] AND ذلك [E] &.

■ *Irrelevant Relation*: if a word has multiple meanings, the constructed pattern may wrongly recognize a relation as causation one. For this kind of error, new patterns need to be added and associated with the void value to exclude the expression that has caused the defect. For instance the word "لذلك" in sentence (5) acts as anaphoric reference. The new pattern (6) indicates an irrelevant indicator.

(5)   اقرأ نشرة الدواء قبل تناول أي جرعة منه فقد لا يكون **لذلك** الدواء أي علاقة بمرضك.
"Read the drug leaflet before taking it since ***that*** drug may not be adequate to your illness"

(6) X   C لذلك DTNN C &.

■ *Misidentify Slots*: in some cases even a relevant relation was correctly extracted, though the patterns failed to fill the slots properly. A good remedy for this defect is to reorder the patterns in a way that more specific patterns have the priority over the more general ones. For example, pattern (8) is not sufficient to correctly fill the *cause* and the *effect* slots of the causal relation in sentence (7); therefore an additional pattern, as the one in (10), needs to be inserted before pattern (8).

(7)   يعاني الميزان التجاري من الخلل **ولذلك** فإن الحكومة بدأت بإقامة المشروعات التي تعتمد على الخدمات
"Trade deficit has prompted the government to develop the services sector"

(8)   R   (&C) [E] لذلك [C] &.
(10) R   (&C) [E] (And) فإن لذلك [C] &.

Patterns were formulated using a series of different kind of tokens separated by space. Tokens comprise the following items:

● *Particular Word:* such as the words "نظرا" in pattern (2).
● *Subpattern Reference:* refers to a list containing a sequence of tokens. For instance the subpattern *&This* in pattern (2) refers to a list of definite demonstrative nouns.
● *Part-of-Speech tag:* represents a certain syntactic category that has been assigned to a text word as the definite noun tag in pattern (6).
● *Slot:* reflects the *cause* or the *effect* part of the relation under scrutiny.
● *Symbol:* instructs the program to take specific action during the pattern matching procedure. For example the plus symbol in pattern (2) instructs the matching procedure to add the word "نظرا" to the *cause* slot of the relation.

4   Experimental Results

The generated patterns were applied to a set of 200 sentences taken from the contemporary Arabic corpus[2] which belong to the science category. Three native Arabic speakers were asked to manually identify the presence of causal relations indicated by causal links in each single sentence. Out of the 107 relations picked out by the subjects the patterns could discover a total of 80 relations giving a **Recall** of 0.75 and **Precision** of 0.77. In reviewing the causal relations missed by the patterns, it turned out that 50% of them were selected by the subjects based on the occurrence of "*causation fa'a*" (فاء السببية) which was not taken into consideration in this study, while the other half was located by causal links not included in the list.

5   Conclusion and Future Work

The purpose of this study was to develop an approach for automatic identification of causal relation in Arabic texts. The method operated well using some of NLP techniques. The extraction system is still being developed as the patterns set has not been completed yet. There are some types of verbs the meaning of which implicitly induce a causal element; these verbs are called *causative verbs* for example "يقتل (kill)", يولد (Generate)" that can be paraphrased as "*to cause to die*" and "*to cause to happen*". Causal relations indicated by the aforementioned types of verbs may be explored in future research.

References

1. Khoo, C.S.G., Kornfilt, J., Oddy, R.N., Myaeng, S.H.: Automatic extracting of cause-effect information from newspaper text without knowledge-based inferencing. Literary and Linguistic Computing 13(4), 177–186 (1988)
2. Carcia, D.: COATIS, an NLP System to Locate Expressions of Actions Connected by Causality Links. In: Plaza, E., Benjamins, R. (eds.) EKAW 1997. LNCS, vol. 1319, pp. 347–352. Springer, Heidelberg (1997)
3. Ittoo, A., Bouma, G.: Extracting Explicit and Implicit Causal Relations from Sparse, Domain-Specific Texts. In: Muñoz, R., Montoyo, A., Métais, E. (eds.) NLDB 2011. LNCS, vol. 6716, pp. 52–63. Springer, Heidelberg (2011)
4. Sadek, J., Chakkour, F., Meziane, F.: Arabic Rhetorical Relations Extraction for Answering "Why" and "How to" Questions. In: Bouma, G., Ittoo, A., Métais, E., Wortmann, H. (eds.) NLDB 2012. LNCS, vol. 7337, pp. 385–390. Springer, Heidelberg (2012)
5. Haskour, N.: Al-Sababieh fe Tarkeb Al-Jumlah Al-Arabih. Aleppo University (1990)

---

[2] `http://www.comp.leeds.ac.uk/eric/latifa/research.htm`

A Framework for Employee Appraisals Based on Inductive Logic Programming and Data Mining Methods

Darah Aqel and Sunil Vadera

School of Computing, Science and Engineering
Newton Building, University of Salford
M5 4WT, Salford, UK
D.M.AbdelrahmanAqel@edu.salford.ac.uk, S.Vadera@salford.ac.uk

Abstract. This paper develops a new semantic framework that supports employee performance appraisals, based on inductive logic programming and data mining techniques. The framework is applied to learn a grammar for writing SMART objectives and provide feedback. The paper concludes with an empirical evaluation of the framework which shows promising results.

1   Introduction

Employee appraisal systems are extensively required for evaluating employee performance [1]. Even though appraisal systems have numerous benefits, some employees question their fairness [2]. The existing commercial systems for appraisals focus on recording information and not supporting goal setting or ensuring that the objectives are SMART (specific, measurable, achievable, realistic, time-related) [3].

Developing a supportive appraisal system for goal setting represents a major challenge. Thus, helping employees to write SMART objectives requires finding the rules of writing the objectives. As the objectives are expressed in natural language, natural language processing (NLP) [4] techniques may have the potential to be used for defining the process of setting SMART objectives. NLP is based on extracting structured information from unstructured text by using automatic methods such as machine learning methods [5]. Inductive Logic Programming (ILP) [6] is a machine learning discipline which extracts rules from examples and background knowledge.

As well as having rules that help structure objectives, there is a need to assess if a stated objective can be met given the available resources and time. Therefore, data mining techniques [7] may have the potential to be used for assessing the objectives.

This paper explores the use of machine learning and data mining techniques for developing a novel system which supports employee appraisals. A new semantic framework for appraisal systems is proposed. The framework facilitates the setting of SMART objectives and providing feedback by using ILP to induce rules that define a grammar for writing SMART objectives. The framework also utilises data mining techniques for assessing the objectives. Parts of the framework have been implemented and an empirical evaluation for the framework has been conducted.

The remaining of the paper is organized as follows. Section 2 proposes the system framework. Section 3 describes the corpus and its tagging. Section 4 describes the use

of ILP for writing SMART objectives. Section 5 illustrates the use of data mining techniques for assessing the objectives**.** Section 6 presents the empirical evaluation of the framework and concludes the paper.

2 The Proposed System Framework

The developed framework for supporting employee appraisals is presented in Fig.1. The framework supports the setting of SMART objectives and providing feedback. The framework uses ILP to learn a grammar for writing SMART objectives in order to ensure that the objectives are "specific", "measurable" and "time-related". The framework utilises data mining methods for assessing whether the objectives are "achievable" and "realistic". Text annotation of the corpus has been done by using GATE [8], WordNet [9] and WordNet SenseRelate AllWords (SR-AW) [10].
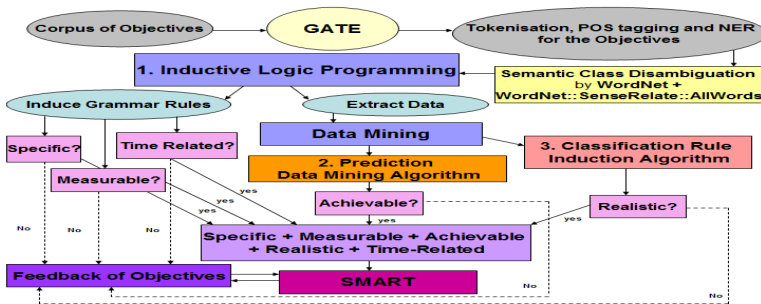
Fig. 1. System Framework

3 The Corpus and Its Tagging

A corpus of objectives has been developed containing 150 example sentences related to the sales domain. This corpus has been created based on studying what constitutes well written objectives and reviewing some SMART objective examples [3].

The GATE system is utilised for annotating the text in the developed corpus. GATE (General Architecture for Text Engineering) is a publicly available system developed at the University of Sheffield for language engineering. Based on GATE, the sentences (objectives) in the developed corpus are first tokenized then the part of speech (POS) annotations and the named entity annotations (NE) are specified.

The semantic tagging is done on the POS-tagged corpus by using WordNet and the SR-AW software to determine the semantic classes of target words (verbs, nouns) that occur frequently in SMART objectives. Results show that the action verbs (e.g. increase, achieve, boost) that are used frequently in writing SMART objectives are classified into one of the following verb semantic classes: "change", "social", "possession", "creation" or "motion". Nouns (e.g. sales) which are used commonly in writing SMART objectives are classified into the noun semantic class "possession".

Some SMART objectives related to different domains (e.g. costs, profits) have been examined semantically as well. Results show that the target words in these objectives are classified into the same classes as the target words in the developed corpus.

To evaluate the accuracy of SR-AW, a corpus that consists of manually annotated objective examples with WordNet semantic classes is used. For a sample of 30 target words, the software has disambiguated 76% of words correctly; where 20% of words have been classified with semantic tagging errors and 4% of them are ambiguous.

4   Using ILP for Writing SMART Objectives

The study uses ILP for writing SMART objectives. ILP uses logic programming and machine learning to induce theories from background knowledge and examples [6]. The inductive machine learning called "ALEPH"[1] is applied on the POS and semantically tagged corpus to learn a grammar for writing SMART objectives in order to ensure that the objectives are "specific", "measurable" and "time-related". An annotated set of 70 sentences is provided to ALEPH as input, together with background knowledge and some examples. The positive (170 examples) and negative (185 examples) example sets have been used for describing the "specific", "measurable" and "time-related" examples. ALEPH has induced 24 linguistic rules for writing SMART objectives. ALEPH has achieved an accuracy of 91% for the training data (proportion: 70%) and 81% for the testing data (proportion: 30%).

ALEPH has induced several rules, including the following PROLOG rule for ensuring that an objective is "specific":

```
specific(A,B) :-
        to(A,C), change_verb (C,D), product(D,E),
        possession_noun_nns(E,F), preposition_in(F,G), percent (H,B).
```

The following PROLOG rule is one of the induced rules by ALEPH for ensuring that an objective is "measurable":

```
measurable(A,B):-
          percent(A,B)
```

The following PROLOG rule is induced by ALEPH for ensuring that an objective is "time-related":

```
time_related(A, B):-
          date(A, B).
```

---

[1] www.cs.ox.ac.uk/activities/machlearn/Aleph/aleph.html

5   Utilising Data Mining Techniques for Assessing the Objectives

For ensuring that the objectives are "achievable", the linear regression algorithm [7] is applied within WEKA data mining tools [7] on a dataset for product sales obtained from the Infochimps repository[2]. The algorithm estimates the expected product sales for a given time. Then the idea is to compare this with the value set in the objective. The achieved accuracy of the model is 98% for mobile data and 91% for PC data.

The rule induction algorithm "JRIP" [7] is applied within WEKA on a randomly selected dataset to ensure that objectives are "realistic". The performance of the model is evaluated using 10 fold cross validation and achieved an accuracy of 95%.

6   Empirical Evaluation and Conclusions

To carry out an empirical evaluation of the framework, a corpus of 150 objective examples (100 SMART, 50 non-SMART) is used. The performance of the system that is based on the rules produced by the ILP and data mining techniques is estimated using 10 fold cross validation. In each validation, 90% is used for training and 10% is used for testing. This results in an outstanding 83% for accuracy. In the future work, more experiments with larger corpus will be carried out and an interactive system that aids employee appraisals and goal setting will be developed.

References

1. Murphy, K., Cleveland, J.: Performance Appraisal: An Organizational Perspective, 3rd edn., 349 pages. Allyn and Bacon (1991)
2. Rowland, C., Hall, R.: Organizational Justice and Performance: is Appraisal Fair? EuroMed Journal of Business 7(3), 280–293 (2012)
3. Hurd, A.R., et al.: Leisure Service Management. Human Kinetics, 386 pages (2008)
4. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing, 680 pages. MIT Press, Cambridge (1999)
5. Alpaydin, E.: Introduction to Machine Learning, 2nd edn., 584 pages. MIT Press (2010)
6. Muggleton, S., De Raedt, L.: Inductive Logic Programming: Theory and Methods. Proceedings of Journal of Logic Programming 19(20), 629–679 (1994)
7. Witten, I., et al.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn., 664 pages. Morgan Kaufmann, Elsevier (2011)
8. Cunningham, H., et al.: Developing Language Processing Components with GATE Version 7 (A User Guide), The University of Sheffield (2012), http://GATE.ac.uk/
9. Miller, G., et al.: Introduction to WordNet: An On-line Lexical Database. International Journal of Lexicography 3(4), 235–244 (1990)
10. Pedersen, T., Kolhatkar, V.: WordNet: SenseRelate:: All Words - A Broad Coverage Word Sense Tagger that Maximizes Semantic Relatedness. In: The 2009 Annual Conference of the North American Chapter of the Assoc. Comp. Lingui., pp. 17–20 (2009)

---

[2]   www.Infochimps.com/datasets/us-consumer-electronics-sales-and-forecasts-2003-to-2007

A Method for Improving Business Intelligence Interpretation through the Use of Semantic Technology

Shane Givens[1], Veda Storey[1], and Vijayan Sugumaran[2,3]

[1] Department of Computer Information Systems, J. Mack Robinson College of Business
Georgia State University
{pgivens3,vstorey}@gsu.edu
[2] Department of Decision and Information Sciences, School of Business Administration
Oakland University
sugumara@oakland.edu
[3] Department of Global Service Management, Sogang Business School
Sogang University, Seoul, South Korea

Abstract. Although business intelligence applications are increasingly important for business operations, the interpretation of results from business intelligent tools relies greatly upon the knowledge and experience of the analyst. This research develops a methodology for capturing knowledge of employees carrying out the interpretation of business intelligence output. The methodology requires the development of targeted ontologies that contribute to analyzing the output. An architecture is presented.

Keywords: Ontology, Semantic Technology, Business Intelligence, Interpretation.

1   Introduction

Business intelligence (BI) encompasses a wide range of business supporting analytical tools to process increasing amounts of corporate data. The interpretation of output produced by BI applications, however, continues to be performed manually, as does the determination of appropriate actions to be taken based upon that output (Seufert et al. 2005). Knowledge surrounding these interpretations and actions builds within the members of an organization. When employees depart an organization, they take that knowledge with them, creating a vacuum.

The objectives of this research are to: define targeted ontologies, and analyze how to use them to support business intelligence initiatives within an organization through the capture of the knowledge of analysts. To do so, a methodology, called the Targeted Ontology for Data Analytics (TODA), is being developed within the context of dynamic capabilities theory. The contribution of the research is to develop an ontology-driven methodology that will facilitate the interpretation of BI results.

2   Research Stages

Stage 1: Targeted ontology development.   Semantic technologies are intended to address questions of interoperability, recognition and representation of terms and

concepts, categorization, and meaning. This research focuses on using ontologies specifically, as a semantic technology that could improve the interpretation of BI application output, especially for novice users.   We develop a targeted ontology to codify the knowledge of the organization being analyzed.   A targeted ontology is defined as: *an ontology that captures and represents organization-specific knowledge in a format that provides direct, formal relationships to the organization's BI environment*. This is done by defining a class of ontology objects called targeted objects which mirror objects in the BI environment, thus, easing the automation of connecting ontological knowledge to BI output.

Stage 2: TODA Artifact Development

This research project follows a design science approach of build and evaluate (Hevner et al. 2004).   The artifact is the TODA methodology, the purpose of which is to provide targeted ontologies that can be applied to improve the interpretation of business intelligence analytics. The TODA methodology consists of four steps: *create targeted ontology, anchor target objects, apply ontological knowledge to BI output, and assess output for new knowledge to be collected*.

Stage 3: Artifact Evaluation

A prototype is being designed that implements the methodology for creating and using targeted ontologies. The TODA Architecture is shown in Figure 1 and consists of: a) user interface, b) ontology creator, c) results interpreter, d) BI environment, and e) external data sources.   The prototype is a necessary tool because refinement of the targeted ontology development is needed.

A Sample Targeted Ontology Scenario

Assume a traditional business intelligence report depicts the results for a query asking for dollar sales of a product category across all stores of a major grocery chain over a period of several weeks. In this case, the category is cookies. Four subcategories of cookies are displayed in the report. For each subcategory, an analyst can discern a pattern of sales over the period of time displayed.   That analyst may be missing key pieces of information. For instance, the analyst may not know there was a stock out of wafer cookies across the company for two weeks in April of 2012. If the company had implemented a targeted ontology, the previous analyst could have input information about the stock out creating a set of nodes as shown below. This would be performed through the ontology creation module of the TODA architecture.

Assume a sample node set with three nodes. The first is the "*Item*" node. This node describes an item involved in the node set. It is also the targeted node. In the physical instantiation of the node, it includes the information necessary (database keys, for example), to tie it to the corresponding information in the organization's business intelligence environment.   The second node in the set describes a stock out that happened to the item in question. It contains detailed information about the stock out. Finally, the third node contains a period of time during which the stock out occurred on the item in question. This node holds the month in this example, but it could hold any period of time.
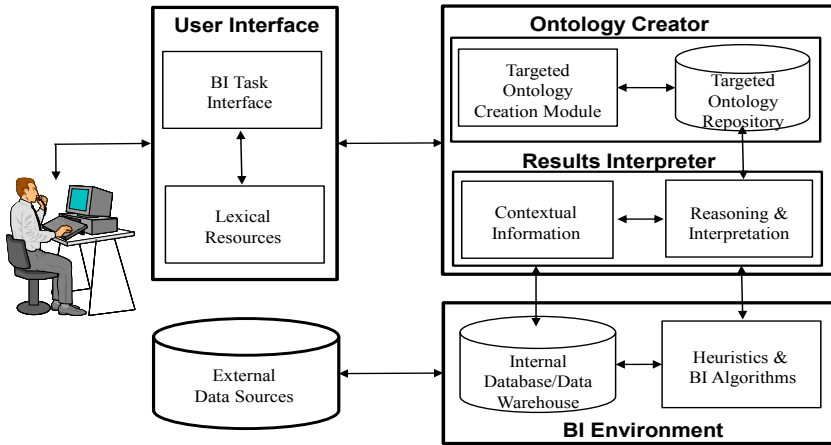
Fig. 1. TODA Architecture

If the organization has implemented the TODA method, the information resulting from the query will be processed through the TODA interpreter (see Figure 1). The interpreter will analyze the information gathered for the report and determine if any targeted nodes exist with links to this data. If so, appropriate notes will be added to the report providing the tacit knowledge from the TODA ontology.

In the original version of the report, there is no contextual information. The analyst only has the numbers provided by the data warehouse to drive any decision-making analysis. In this instance, she may see the dip in sales of wafer cookies in mid-April as a sign of normal seasonality. Maybe there was a build up of cookie demand leading into Easter and, once that was over, sales dipped. This conclusion would be a guess, but given the data she has to examine, there is little else to guide her analysis. The new version of the report results from the data being processed through a TODA-based system. The data from the organization's data warehouse is pre-processed by the TODA results interpreter. The interpreter determines if any values in the data belong to a dimension linked to the targeted ontology through a linked node. If so, it finds any contextual knowledge stored in those nodes and adds it to the report. The report production process is then completed.

The analyst now has access to contextual information about the time period where wafer cookie sales dropped off. In her previous analysis, she determined this was normal seasonality. Now she can see that this was due to a supply disruption. If the analyst was supporting inventory-planning decisions, her previous analysis may have led her to reduce inventories in mid-April. This would have caused stock-outs all over again. Now she has the knowledge necessary to see that more inventory is needed during this period, not less. This is the real contribution of TODA to practice; providing business intelligence analysts with the contextual knowledge needed to make better decisions. For the assessment, the hypotheses to be tested are summarized in Table 2.

Table 1. Hypotheses

H1: Targeted ontologies can improve the performance of BI analysts who are new to an organization
H2: Targeted ontologies can facilitate analysts providing better interpretations of BI output
H3: Targeted ontologies can prevent analysts from misinterpreting BI output

3   Conclusion

This research proposes the use of targeted ontologies to improve the interpretation of business intelligence data. Challenges include developing good procedures and heuristics for eliciting targeted ontologies and then creating the techniques and algorithms needed to effectively apply them. Further work will be needed for additional validation on multiple sites and the inclusion of other semantic technologies.

References

Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 75–105 (2004)

Seufert, A., Schiefer, J.: Enhanced business intelligence-supporting business processes with real-time business analytics, pp. 919–925. IEEE (2005)

Staab, S., Gómez-Pérez, A., Daelemana, W., Reinberger, M.L., Noy, N.: Why evaluate ontology technologies? Because it works! IEEE Intelligent Systems 19(4), 74–81 (2004)

Code Switch Point Detection in Arabic

Heba Elfardy[1], Mohamed Al-Badrashiny[1], and Mona Diab[2]

[1] Columbia University
[2] The George Washington University
heba@cs.columbia.edu,
badrashiny@ccls.columbia.edu,
mtdiab@gwu.edu

Abstract. This paper introduces a dual-mode stochastic system to automatically identify linguistic code switch points in Arabic. The first of these modes determines the most likely word tag (i.e. dialect or modern standard Arabic) by choosing the sequence of Arabic word tags with maximum marginal probability via lattice search and 5-gram probability estimation. When words are out of vocabulary, the system switches to the second mode which uses a dialectal Arabic (DA) and modern standard Arabic (MSA) morphological analyzer. If the OOV word is analyzable using the DA morphological analyzer only, it is tagged as "DA", if it is analyzable using the "MSA" morphological analyzer only, it is tagged as MSA, otherwise if analyzable using both of them, then it is tagged as "both". The system yields an $F_{\beta=1}$ score of 76.9% on the development dataset and 76.5% on the held-out test dataset, both judged against human-annotated Egyptian forum data.

Keywords: Linguistic Code Switching, Diglossia, Language Modeling, Arabic, Dialectal Arabic Identification.

1 Introduction

Linguistic code switching (LCS) refers to the use of more than one language in the same conversation, either inter-utterance or intra-utterance. LCS is pervasively present in informal written genres such as social media. The phenomenon is even more pronounced in diglossic languages like Arabic in which two forms of the language co-exist. Identifying LCS in this case is more subtle in particular in the intra-utterance setting.[1] This paper aims to tackle the problem of code-switch point (CSP) detection in a given Arabic sentence. A language-modeling (LM) based approach is presented for the automatic identification of CSP in a hybrid text of modern standard Arabic (MSA) and Egyptian dialect (EDA) text. We examine the effect of varying the size of the LM as well as measuring the impact of using a morphological analyzer on the performance. The results are compared against our previous work [4]. The current system outperforms our previous implementation by a significant margin of an absolute 4.4% improvement, with an $F_{\beta=1}$ score of 76.5% compared to 72.1%.

---

[1] For a literature review, we direct the reader to our COLING 2012 paper [4].

2   Approach

The hybrid system that is introduced here uses a LM with a back off to a morphological analyzer (MA) to handle out of vocabulary (OOV) words to automatically identify the CSP in Arabic utterances. While the MA approach achieves a far better coverage of the words in a highly derivative and inflective language such as Arabic, it is not able to take context into consideration. On the other hand, LMs yield better disambiguation results because they model context in the process.

2.1   Language Model

The system uses the MSA and EDA web-log corpora from the Linguistic Data Consortium (LDC) to build the language models. [2]Half of the tokens in the language model come from MSA corpora while the other half come from EDA corpora. The prior probabilities of each MSA and EDA word are calculated based on their frequency in the MSA and DA corpora, respectively. For example, the EDA word *ktyr*,[3] meaning *much*, will have a probability of 0 for being tagged as *MSA* since it would not occur in the MSA corpora, and a probability of 1 for being tagged as *EDA*. Other words can have different probabilities depending on their unigram frequencies in both corpora.

All tokens in the MSA corpora are then tagged as *MSA* and all those in the EDA corpora as *EDA*. Using SRILM [7] and the tagged datasets, a 5-gram LM is built with a modified Kneser-Ney discounting.

The LM and the prior probabilities are used as inputs to SRILM's *disambig* utility which uses them on a given untagged sentence to perform a lattice search and return the best sequence of tags for the given sentence.

2.2   Morphological Analyzer

All OOVs are run through CALIMA [5], an MSA and EDA morphological analyzer based on the both the SAMA [6] MA and database as well as the Tharwa three way MSA-EDA-ENG dictionary [2]. CALIMA returns all MSA and EDA analyses for a given word. The OOV word is tagged as "both" if it has MSA and EDA analyses. While it is tagged as "MSA" or "EDA" if it has only MSA or EDA analyses, respectively.

3   Evaluation Dataset

We use three different sources of web-log data to create our evaluation dataset. The first of which comes from the Arabic Online Commentary dataset that was

---

[2] The LDC numbers of these corpora are 2006{E39, E44, E94, G05, G09, G10}, 2008{E42, E61, E62, G05}, 2009{E08, E108, E114, E72, G01}, 2010{T17, T21, T23}, 2011{T03}, 2012{E107, E19, E30, E51, E54, E75, E89, E94, E98, E99}.

[3] We use Buckwalter transliteration scheme,
http://www.qamus.org/transliteration.htm

produced by [8] and consists of user commentaries from an Egyptian newspaper while the second one was crawled from Egyptian discussion forums for the CO-LABA project [1] and finally the third one comes from one of the LDC corpora that are used to build the EDA language model. All datasets are manually annotated by a trained linguist using a variant of the guidelines that are described in [3]. In this variant of the guidelines, the annotation is purely contextual, so that if a word is used with the same sense in MSA and EDA, its label is determined based on the context it occurs in. In rare cases, where enough context is not present, a *both* class is used indicating that the word could be both MSA and EDA. Since we are not currently targeting romanized-text and named-entity identification, we exclude all entries that are labeled as Foreign, or Named-Entity from our evaluation, which correspond to a total of 8.4% of our dataset. Moreover, we also exclude unknown words and typos which only represent 0.7% of our dataset. We split our dataset into a development set for tuning and a held-out set for testing. The development-set has 19,954 MSA tokens (7,748 types), 9,771 EDA tokens (4,379 types) and 9 Both tokens (9 types). The test-set comprises 15,462 MSA tokens (6,887 types), 16,242 EDA tokens (6,151 types) and 5 Both tokens (5 types).

4   Experimental Results

We investigate two experimental conditions: one with the morphological analyzer as a back off turned on, the second mode has the morphological analyzer turned off. Both conditions experiment with varying the size of the LM as follows: 2, 4, 8, 16, 28M words, respectively. We employ two baselines: MAJB, a majority baseline that tags all words with the most frequent tag in our data set; the second baseline, COLB, is the approach presented in [4] using the same datasets that we used in building our language models. Figure 1 shows the $F_{\beta=1}$ of both sets of experiments against the baselines. Our approach significantly outperforms both baselines. One surprising observation is that the $F_{\beta=1}$ decreases as the size of the LM increases beyond 4 million tokens (with a slight drop at the 8M mark). We surmise that this is because as the size of the language model increases, the shared ngrams between MSA and EDA increases. For example, for the 4M LM (where we note the highest $F_{\beta=1}$ score), the shared types represent 21.2% while for the largest LM of size 28M, the shared types represent 27.6% . This causes more confusability between the classes for larger LMs which explains the lower $F_{\beta=1}$ scores despite the higher coverage.

As expected backing-off to the morphological analyzer improves the results especially for the smaller LMs where there is less coverage. However as the size of the LM increases, the coverage increases and the percentage of OOV decreases hence the morphological analyzer becomes less useful. For example, the percentage of OOVs for the 4M LM (when not backing-off to the morphological analyzer) is 7.2% while for the 28M LM it is 3.1%.

On the test set, the system outperforms both baselines with an $F_{\beta=1}$ score of 76.5% using the best configuration (4M tokens with back off to the morphological
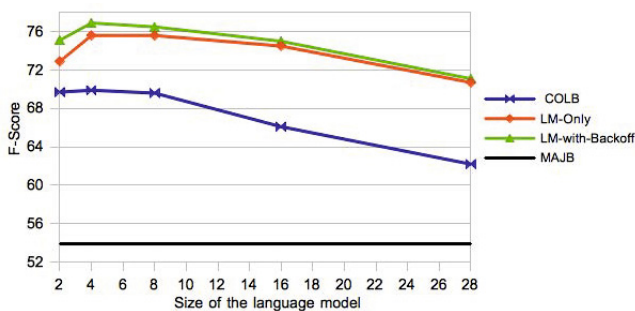
Fig. 1. Weighted Average of F-Scores of the MSA and DA classes with different experimental setups against the baseline systems, MAJB and COLB

analyzer) compared to 34.7% for the majority baseline MAJB and 72.1% for our high baseline system, COLB.

5   Conclusion

We presented a new dual-mode stochastic system to automatically perform point-level identification of linguistic code switches in Arabic. We studied the impact of varying the size of the language model with and without employing a morphological analyzer as a back-off method to handle the OOV. Our best (using the LM plus the morphological analyzer as a back-off) system achieves an F-Score of 76.9% and 76.5% on the development and test datasets, respectively. These results outperformed both the majority baseline and our previous approach introduced in [4].

References

1. Diab, M., Habash, N., Rambow, O., Altantawy, M., Benajiba, Y.: Colaba: Arabic dialect annotation and processing. In: LREC Workshop on Semitic Language Processing, pp. 66–74 (2010)
2. Diab, M., Hawwari, A., Elfardy, H., Dasigi, P., Al-Badrashiny, M., Eskandar, R., Habash, N.: Tharwa: A multi-dialectal multi-lingual machine readable dictionary (forthcoming, 2013)
3. Elfardy, H., Diab, M.: Simplified guidelines for the creation of large scale dialectal arabic annotations. In: LREC, Istanbul, Turkey (2012)

4. Elfardy, H., Diab, M.: Token level identification of linguistic code switching. In: COLING, Mumbai, India (2012)
5. Habash, N., Eskander, R., Hawwari, A.: A Morphological Analyzer for Egyptian Arabic. In: NAACL-HLT Workshop on Computational Morphology and Phonology (2012)
6. Maamouri, M., Graff, D., Bouziri, B., Krouna, S., Bies, A., Kulick, S.: Ldc standard arabic morphological analyzer (sama) version 3.1 (2010)
7. Stolcke, A.: Srilm an extensible language modeling toolkit. In: ICSLP (2002)
8. Zaidan, O.F., Callison-Burch, C.: The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content. In: ACL (2011)

SurveyCoder: A System for Classification of Survey Responses

Sangameshwar Patil* and Girish K. Palshikar

Tata Research Development and Design Centre, TCS
54B, Hadapsar Industrial Estate, Pune 411013, India

Abstract. Survey coding is the process of analyzing text responses to open-ended questions in surveys. We present SurveyCoder, a research prototype which helps the survey analysts to achieve significant automation of the survey coding process. SurveyCoder's multi-label text classification algorithm makes use of a knowledge base that consists of linguistic resources, historical data, domain specific rules and constraints. Our method is applicable to surveys carried out in different domains.

Keywords: Text classification, Survey analytics, NLP.

1 Introduction

Open-ended questions are a vital component of a survey as they elicit subjective feedback. Data available from responses to open-ended questions has been found to be a rich source for variety of purposes. However, the benefits of open-ended questions can be realized only when the unstructured, free-form answers which are available in a natural language (such as English, German, Hindi and so on.) are converted to a form that is amenable to analysis.

Survey coding is the process that converts the qualitative input available from the responses to open-ended questions to a quantitative format that helps in quick analysis of such responses. The set of customer responses in electronic text format (also known as *verbatims*) and a pre-specified set of codes, called code-frame constitute the input data to the survey-coding process. A code-frame consists of a set of tuples (called code or label) of the form <code-id, code-description>. Each code-id is a unique identifier assigned to a code and the code-description usually consists of a short description that "explains" the code. Survey coding task is to assign one or more of codes from the given code-frame to each customer response. As per the current practice in market research industry, it is carried out by specially trained human annotators (also known as coders). Sample output of the survey-coding process is shown in Fig. 1.

The research community has approached the problem of automatic code assignment from multiple perspectives. An active research group in this area is led by Sebastiani and Esuli et al. [1–3]. They approach the multiclass coding problem using a combination of active learning and supervised learning. Almost

---

* Sangameshwar Patil is also a doctoral research scholar at Dept. of CSE, IIT Madras.

Verbatim 1: ease of use....covers allergens as well	*Verbatim 2:* because it is cheap and I like the scentts
`05: easy to use`	`08: like fragrance`
`03: removes allergens`	`01: good/reasonable price`

Fig. 1. Output of survey coding: Examples of verbatims and codes assigned to them

all the supervised learning techniques mentioned in the current literature need training data which is specific to each survey. This training data is not available with the survey and has to be created by the human annotators to begin with. In most of the cases, the cost and effort required to create necessary training data outweighs the benefits of using supervised learning techniques. Thus use of supervised learning techniques is not the best possible solution.

2   Our Approach : SurveyCoder

We have implemented a working prototype of SurveyCoder to automate the survey coding process. SurveyCoder follows a two-stage approach and uses unsupervised text classification and active learning techniques. It makes use of a knowledge base that captures the historical survey coding data, domain specific knowledge and constraints as well as linguistic resources such as the WordNet [4].

Pre-processing Stage: First, the input data is passed through a data-cleansing component that carries out spelling correction, acronym expansion etc. The second step uses natural language processing technique known as word sense disambiguation (WSD) to identify the likely senses of words in the input text. We use WordNet synsets (and corresponding sense numbers) to distinguish between multiple meanings of a given word. The historical knowledge base along with the survey question and survey domain are significantly useful for this task. In the third step, the feature extractor module represents each code and verbatim in terms of semantic units (SemUnit) which will be used by the classifier. Purpose of this module is to attach a weight (a fractional number as measure of relative importance of the word with respect to other words in code description) and represent each word in terms of its semantics and to capture the concept expressed in a code or a verbatim. We represent each word using its part-of-speech tags as well as its WordNet sense ids. For a given word, this enables us to find out synonyms, antonyms as well as other related words (hypernyms, holonyms, etc.).

As an illustrative example, consider representation of a sample code at the end of pre-processing stage; `c1: relieves#v#1,4,7#i2 pain#n#1#i2`. This representation of code c1 denotes that out of all available WordNet senses of that word, we consider sense numbers 1, 4 and 7 for the word "relieves" with its part-of-speech tag as verb. Further, we use one of four pre-determined weights (fractional numbers) to capture the relative importance of each word in a particular codes description. These weights are denoted using following labels:

- **i0** = 0.0 : a word with the importance **i0** is not important at all.
- **i1** = 0.64 : most important word for that particular code and will cause the code to be assigned to a verbatim containing this word in the first round of code assignment. (Note that in subsequent rounds of assignment and post-processing logic, this assignment may get modified.)
- **i2** = 0.4 : needs to be combined with another word from code description which has importance of **i2** or higher.
- **i3** = 0.3 : this word must be combined with at least two other words from code description which have importance of **i3** or higher to cause code assignment.

Algorithm AssignCodes

Input: code frame **F** = {(a_1, C_1), (a_2, C_2), ..., (a_M, C_M)} // a_i = code-ID, C_i = textual code description

Input: set of documents **D** = {$D_1, D_2, ..., D_N$} // each document D_i is an ordered list of 1 or more sentences

Output: {(D_1, L_1), (D_2, L_2), ..., (D_N, L_N)} // a subset $L_i \subseteq \{a_1, a_2, ..., a_M\}$ of code-IDs assigned to each document $D_i \in$ **D**

1. Find out overlap between the semantic unit based representation of each code and the words in each sentence for each document. Each overlapping word is weighted with the importance of the word in the code description and represents the belief that this particular code is applicable to the given document D_i. Multiple such beliefs are combined using certainty algebra to find a single value for the belief that a particular code is applicable to a document D_i. Assign a subset of labels $L_i \subseteq \{a_1, a_2, ..., a_M\}$ of code-IDs to each document $D_i \in$ **D** for which the belief is above the threshold **θ**.

2. Human coder offers corrections to the code-IDs assigned to a subset of the documents in **D** (this subset is either selected by *SurveyCoder* or by the coder himself).

   If there are no corrections, **then stop. If** human coder wants to stop **then stop.**

3. Refine the importance of word senses for the codes for which the corrections were offered by the human coder.

4. If the number of iterations (or corrections) is more than a pre-set limit **then stop else** go to step 2.

Fig. 2. AssignCodes algorithm

Code-Assignment Stage: In the code-assignment stage (Fig. 2), we make use of the semantic unit based representation of each code to find out overlap between that code's textual description and the words in each sentence for each document. We group this lexical overlap along five major word categories viz. nouns, verbs, adjectives, adverbs and cardinals. Each overlapping word is weighted with the importance of the word in the code description and is used to quantify our belief about whether the corresponding code can be assigned to the given document D_i. To decide whether a code is applicable to a document, we need to combine the evidence presented by multiple overlapping words. For this purpose, we use the certainty factor algebra [5] to get a single, consolidated value for this belief. If this belief is above certain threshold (denoted by Θ), we assign the code to the document. Based on the carefully chosen values for the importance factors (**i0, i1, i2, i3**) as described in previous section, the value of this threshold Θ is chosen to be 0.6.

3   Experimental Results

We have evaluated SurveyCoder using multiple survey datasets from diverse domains such as over-the-counter-medicine, household consumer goods (e.g. detergents, fabric softners etc.), food and snack items, customer satisfaction surveys, campus recruitment test feedback surveys etc. Fig. 3 summarizes some of our results for classification of survey responses (without using any feedback).

Survey_Category	Number of codes in the code-frame	Number of respondents	Recall	Precision	F-1 measure
OTC_Meds (Female medicines)	83	256	86.05	81.82	83.88
OTC_Meds (Common cold / Allergy)	103	538	79.38	74.97	77.11
Household goods (Shaving)	140	763	82.64	70.15	75.88
Food-items (Snack-Baked_Goods)	136	196	79.94	71.39	75.42
Pet items (Dog_Food)	107	1153	74.50	68.10	71.15

Fig. 3. Sample results for classification of survey responses in diverse domains

4   Conclusion

In this paper, we presented SurveyCoder, a research prototype which helps the survey analysts to achieve significant automation of the survey coding process. We have observed that our minimally supervised approach to classifying the survey responses works reasonably well for analysing surveys for diverse domains. As part of further work, we are working on improving the accuracy of SurveyCoder to incorporate the human feedback using active learning techniques.

References

1. Giorgetti, D., Sebastiani, F.: Multiclass text categorization for automated survey coding. In: Proceedings of ACM Symposium on Applied Computing (SAC) (2003)
2. Esuli, A., Sebastiani, F.: Active learning strategies for multi-label text classification. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 102–113. Springer, Heidelberg (2009)
3. Esuli, A., Sebastiani, F.: Machines that learn how to code open-ended survey data. International Journal of Market Research 52(6) (2010)
4. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)
5. Buchanan, B., Shortliffe, E.: Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley, Reading (1984) ISBN 978-0-201-10172-0

Rhetorical Representation and Vector Representation in Summarizing Arabic Text

Ahmed Ibrahim and Tarek Elghazaly

Department of Computer and Information Sciences,
Institute of Statistical Studies and Research, Cairo University, Egypt
a.ibr@live.com, tarek.elghazaly@cu.edu.eg

Abstract. This paper examines the benefits of both the Rhetorical Representation and Vector Representation for Arabic text summarization. The Rhetorical Representation uses the Rhetorical Structure Theory (RST) for building the Rhetorical Structure Tree (RS-Tree) and extracts the most significant paragraphs as a summary. On the other hand, the Vector Representation uses a cosine similarity measure for ranking and extracting the most significant paragraphs as a summary. The framework evaluates both summaries using precision. Statistical results show that Rhetorical Representation is superior to Vector Representation. Moreover, the rhetorical summary keeps the text in context, without leading to lack of cohesion in which the anaphoric reference is not broken i.e. improving the ability of extracting the semantics behind the text.

Keywords: Rhetorical Structure Theory, RST, Arabic text summarization, Rhetorical Representation, Vector Representation, RS-Tree, Cosine Similarity.

1 Introduction and Previous Work

Automatic Text Summarization is one of the most difficult problems in Natural Language Processing (NLP) [1]. This paper presents a framework using two summarization techniques and evaluating the output summary with manual summary. Recent researches use different methods for Arabic text summarization such as: text structure and topic identification [1], linguistic using RST [2,3], machine learning technique [1,5].

2 The Proposed Framework

The Proposed framework uses the two summarization techniques: Rhetorical Representation based on RST as it is in [3,4] and Vector Representation based on Vector Space Model (VSM). It also examines the pros and cons of both summaries by extracting the most significant paragraphs of the original text as shown in the following Fig. (1).

The framework starts with preparing a test set. It is extracted using the Really Simple Syndication (RSS) (which is called a "feed", "web feed" or "channel"). In this paper, the RSS reader uses the BBC's online Arabic news portal to extract fully news

articles. Articles include different kinds of news including, general, political, business, regional, and entertainment news. The average paragraphs of the article are five and the average words in each paragraph are 24 words. The test set is classified into three groups: small-sized articles (1-10 paragraphs), medium-sized articles (11-20 paragraphs), and large-sized articles (21-40 paragraphs). The overall figures of the test set are illustrated in Table (1).
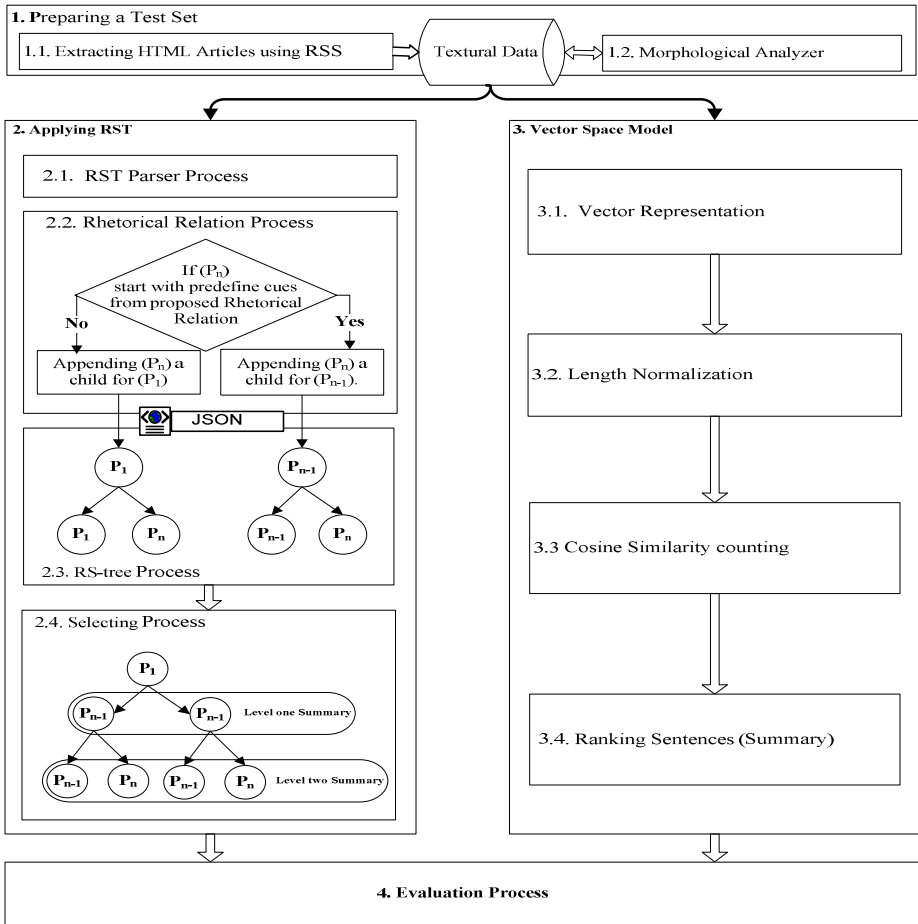
Fig. 1. A proposed framework

The proposed framework is applied to the RST as shown in Fig. (1) through four steps. First, the original input text is segmented into paragraphs (as indicated by the HTML <p> tags). Second, paragraphs are classified into the nucleus or satellites depend on the algorithm in [3,4], and a JSON code (JavaScript Object Notation) is produced. Third, a text structure is represented by using the JSON code and the RS-Tree is built. Finally, the nucleus nodes (significant paragraphs) are selected from the RS - Tree.

Table 1. Statistics the Test Set

Category	Figures
Corpus Textual Size	25.06 MB
No. of Articles	212
No. of Paragraphs	2249
No. of Sentences	2521
No. of words (exact)	66448
No of Word (root)	41260
No. of Stop word	15673
No. of small-sized articles (Less than 10 paragraphs)	104
No. of medium-sized articles (10 - 20 paragraphs)	79
No. of large-sized articles (More than 20 paragraphs)	29

The proposed framework also, applies the VSM as shown in Fig. (1) by representing the article parts (title, paragraphs) as vectors; and computing the cosine similarities for each paragraph vector based on resemblance to the title vector. Furthermore, for scoring comparable weights, long and short paragraphs should be normalized by dividing each of their components according to the length. Computing the cosine similarities uses the following equation and selects the top [6].

$$Cosine\ Similarity(V_t, V_p) = \frac{V_t \cdot V_p}{|V_t|\ |V_p|} = \frac{V_t}{|V_t|} \cdot \frac{V_p}{|V_p|} = \frac{\sum_{i=1}^{|V|} t_i\ p_i}{\sqrt{\sum_{i=1}^{|V|} (t_i)^2}\ \sqrt{\sum_{i=1}^{|V|} (p_i)^2}}$$

Where:

V_t is the *tf•idf* weight of term t_i in the title.

V_p is the *tf•idf* weight of term t_i in the paragraph

The tf•idf vector is composed of the product of a term frequency and the inverse document frequency for each title terms that appears in the all article paragraphs.

3   Experiments and Results

Fig. (2) clarifies the precision which is based on the result of experiments. The Y-axis represents the precision, and the X-axis represents the text size groups. VSM-summary achieves average precision of 53.13%; whereas, RST-summary achieves 56.29%. However, in the large-sized articles the VSM-summary precision achieves 42.7% more than RST-summary which achieves only 39.02%. At the quality of summary in itself (intrinsic), the RST-summary has kept result not out of context, without lacking of cohesion and anaphoric reference not broken.
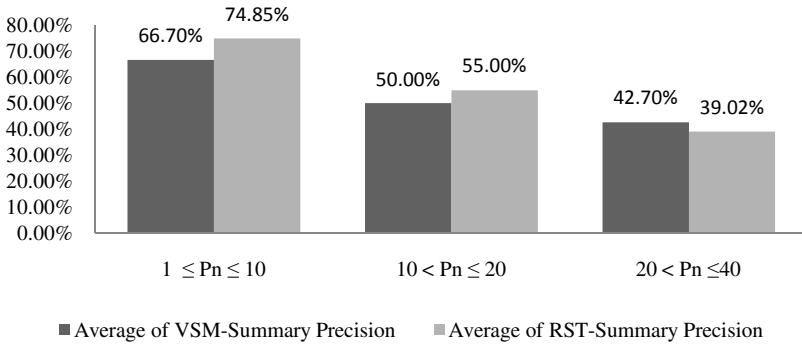
Fig. 2. Performance of both *VSM* and *RST* summary results with the judgments

4   Conclusion and Future Works

RST is a very effective technique for extracting the most significant text parts. However, the limitation of RST appears when it is applied to large-sized articles. Statistical results show that RST-summary is superior to VSM-summary: the average precision for RST-summary is 56.29%; whereas, that of VSM-summary is 53.13%. Besides, the VSM-Summary is incoherent and deviates from the context of the original text.

In the Future works, these two models may be combined together to provide a new model that will improve the summary results by inlaying the rhetorical structure tree using weights of VSM-summary. In addition, RS-Tree can be used to identify the written styles of different writers.

References

1. Hammo, B.H., Abu-Salem, H., Martha, E.W.: A Hybrid Arabic Text Summarization Technique Based on Text Structure and Topic Identification. Int. J. Comput. Proc. Oriental Lang. (2011)
2. Alsanie, W., Touir, A., Mathkour, H.: Towards an infrastructure for Arabic text summarization using rhetorical structure theory. M.Sc. Thesis, King Saud University, Riyadh, Saudi Arabia (2005)
3. Ibrahim, A., Elghazaly, T.: Arabic text summarization using Rhetorical Structure Theory. In: 8th International Conference on Informatics and Systems (INFOS), pp. NLP-34–NLP-38 (2012)
4. Ibrahim, A., Elghazaly, T.: Rhetorical Representation for Arabic Text. In: ISSR Annual Conference the 46th Annual Conference on Statistics, Computer Science, and Operations Research (2011)
5. Abd-Elfattah, M., Fuji, R.: Automatic text summarization. In: Proceeding of World Academy of Science, Engineering and Technology, Cairo, Egypt, pp. 192–195 (2008)
6. Manning, C., Raghavan, P., Schütze, H.: An Introduction to Information Retrieval, p. 181. Cambridge University Press (2009)

Author Index