

Proceedings of the
International Workshop & Tutorial on
Adaptive Text Extraction and Mining

held in conjunction with the 14th European
Conference on Machine Learning and the 7th
European Conference on Principles and Practice of
Knowledge Discovery in Databases

22 September 2003
Cavtat–Dubrovnik (Croatia)

`www.dcs.shef.ac.uk/~fabio/ATEM03`

Organizing Committee:

Fabio Ciravenga (University of Sheffield, UK)
Nicholas Kushmerick (University College Dublin, Ireland)

Table of Contents

Preface	1
Exploiting the feature vector model for learning linguistic representations of relational concepts (<i>Roberto Basili, Maria Teresa Pazienza, Fabio Massimo Zanzotto</i>)	2
Automatic acquisition of taxonomies from text: FCA meets NLP (<i>Philipp Cimiano, Steffen Staab, Julien Tane</i>)	10
Active learning selection strategies for information extraction (<i>Aidan Finn, Nicholas Kushmerick</i>)	18
Active learning for information extraction with multiple view feature sets (<i>Rosie Jones, Rayid Ghani, Tom Mitchell, Ellen Riloff</i>)	26
Information extraction from multi-document threads (<i>David Masterson, Nicholas Kushmerick</i>)	34
An analysis of ontology-based query expansion strategies (<i>Roberto Navigli, Paola Velardi</i>)	42
Combining ontological knowledge and wrapper induction techniques into an e-retail system (<i>Maria Teresa Pazienza, Armando Stellato, Michele Vindigni</i>)	50
Meta-learning beyond classification: A framework for information extraction from the Web (<i>Georgios Sigletos, Georgios Paliouras, Constantine D. Spyropoulos, Takis Stamatopoulos</i>)	58
Information extraction via double classification (<i>An De Sitter, Walter Daelemans</i>)	66
Information extraction as a Semantic Web technology: Requirements and promises (<i>Mark Stevenson, Fabio Ciravegna</i>)	74
Finding educational resources on the Web: Exploiting automatic extraction of metadata (<i>Cynthia Thompson, Joseph Smarr, Huy Nguyen, Christopher Manning</i>)	79
Semantic relations in concept-based cross-language medical information retrieval (<i>Špela Vintar, Paul Buitelaar, Martin Volk</i>)	83
Tutorial Notes (<i>Fabio Ciravenga, Nicholas Kushmerick</i>)	92

Preface

Vast quantities of valuable knowledge are embedded in unstructured textual formats. Petabytes of text are currently available on the public Web, in intranets and other private repositories, and on our personal desktop machines. In many cases, the only way to access such documents is through blunt instruments such as keyword-based document retrieval. In recent years, there has been significant research (and considerable commercial) interest in technologies for automatically extracting and mining useful structured knowledge from unstructured text. Current trends suggest a movement away from pure natural language processing approaches requiring the manual development of rules, to shallower, less knowledge-intensive approaches based on techniques from machine learning, information retrieval and data mining.

Adaptive text extraction and mining is an enabling technology with a wide variety of applications. On the Web, automated knowledge capture from text would open the way for both better retrieval, and advanced business applications (e.g. B2B/B2C applications mediated by knowledge-aware agents). For knowledge management, capturing the knowledge contained in a company's repositories would encourage knowledge to be shared and reused among employees, improving efficiency and competitiveness. Extracting information from texts is an important step in capturing knowledge, e.g. for populating databases or ontologies, supporting document annotation (e.g. for the Semantic Web), for learning ontologies, etc.

Following the tradition of the previous workshops on the same topic held at AAAI-1998 (www.isi.edu/info-agents/RISE/ML4IE), ECAI-2000 (www.dcs.shef.ac.uk/~fabio/Local/ecai-workshop.html), and IJCAI-2001 (www.smi.ucd.ie/ATEM2001), ATEM-03 will bring together researchers and practitioners from different communities (e.g. machine learning, text mining, natural language processing, information extraction, information retrieval, ontology learning), to discuss recent results and trends in mining texts for knowledge capture.

The program includes two types of submissions: nine long papers describing completed research, and three short papers describing ongoing work or challenging ideas. Ten papers come from Europe (Belgium, Germany, Greece, Italy, Ireland, United Kingdom), two come from USA. Each paper was reviewed by at least two reviewers. Acceptance rate was 46% (12 out of 26).

A short tutorial is associated with the workshop, providing an introduction to Information Extraction from Web Documents. This tutorial follows on an earlier successful tutorial at the European Conference on Artificial Intelligence in 2002 in Nantes.

We thank the reviewers for their cooperation in the reviewing process, especially considering that the large number of submissions caused a heavier load than expected. The Programme Committee was comprised of:

Christopher Brewster	University of Sheffield, UK
Joe Carthy	University College Dublin, Ireland
Philipp Cimian	University of Karlsruhe, Germany
Valter Crescenzi	Università di Roma Tre, Italy
An De Sitter	University of Antwerp, Belgium
Dayne Freitag	Fair Isaac Corporation, USA
Rayid Ghani	Accenture Technology Labs, USA
Rosie Jones	Accenture Technology Labs, USA
Christopher Manning	Stanford University, USA
Ion Muslea	SRI International, USA
Hwee Tou Ng	National University of Singapore, Singapore
Horacio Saggion	University of Sheffield, UK
Mark Stevenson	University of Sheffield, UK

Exploiting the feature vector model for learning linguistic representations of relational concepts

Roberto Basili, Maria Teresa Pazienza, Fabio Massimo Zanzotto

University of Rome "Tor Vergata",

Department of Computer Science, Systems and Production,

00133 Roma (Italy)

{basili, pazienza, zanzotto}@info.uniroma2.it

Abstract

In this paper we focus our attention to the construction of one-to-many mappings between the coarse-grained relational concepts and the corresponding linguistic realisations with an eye on the problem of selecting the catalogue of the coarse-grained relational concepts. We here explore the extent and nature of the general semantic knowledge required for the task, and, consequently, the usability of general-purpose resources such as WordNet. We propose an original model, the verb semantic prints, for exploiting ambiguous semantic information within the feature vector model.

1 Introduction

Relational concepts and their linguistic realisations are very relevant bits of semantic dictionaries. These equivalence classes, often called *semantic frames*, may enable sophisticated natural language processing applications as argued in [7] among others. For example, take the relational concept *have-revenues*(AGENT:*X*, AMOUNT:*Y*, TIME:*Z*) and two related "generalised" forms *X has a positive net income of Y in Z* and *X reports revenues of Y for Z*. This would help in finding answers to very specific factoid questions such as "*Which company had a positive net income in the financial year 2001?*" using text fragments as "*Acme Inc. reported revenues of \$.9 million for the year ended in December 2001.*".

Information Extraction (IE) is based on this notion. Templates are relational concepts and extraction patterns are linguistic realisations of templates or, eventually, of intermediate relational concepts, i.e. the events. Besides used techniques we can say that IE is a *semantic-oriented* application.

Generally, such a kind of applications rely on complete semantic models consisting of: a catalogue of

named entity classes (relevant concepts) as *Company*, *Currency*, and *TimePeriod*; a catalogue of (generally) coarse-grained relational concepts with their semantic restrictions, e.g. *have-revenues*(AGENT:*Company*, AMOUNT:*Currency*, TIME:*TimePeriod*); a set of rules for detecting named entities realised in texts and assigning them to the correct class; and, finally, a catalogue of one-to-many mappings between the coarse-grained relational concepts and the corresponding linguistic realisations. These semantic models are often organised using logical formalisms (as in [6]). The results are very interesting artifacts conceived to represent equivalences among linguistic forms in a systematic and principled manner.

Besides the representational formalism, the actual content of semantic models is a crucial issue. Using *semantic-oriented* systems requires the definition of the relevant semantic classes and their one-to-many mappings with the linguistic realisations within the target knowledge domain. Even if repositories of general knowledge about the world exist both at the concept level (e.g. Wordnet [11]) and at the relational concept level (e.g. Framenet [2]), they can be hardly straightforwardly used. Specific domains and information needs such as airplane travels in [16] or the company mergers and acquisitions in [1] generally stress their limits. Good coverage of phenomena and, consequently, good performances of final applications can be reached when the underlying semantic models are adapted to target domains.

It is reasonable to hope that the cost of building domain-specific semantic resources can be dramatically reduced as such a kind of knowledge already exists in "natural" repositories: the domain corpora. We are interested in investigating this problem relying on a "terminological" perspective [5]. It is our opinion that typical insights of terminology studies as *admissible surface forms* and *domain relevance* help in concentrating the attention on relevant and generalised text fragments when mining large text collections.

In this paper we focus our attention to the construction of one-to-many mappings between the coarse-grained relational concepts and the corresponding linguistic realisations with an eye on the problem of selecting the catalogue of the coarse-grained relational concepts. As it will be clarified in Sec. 2 that describe the terminological approach, we work on a list of extraction patterns derived from the analysis of the domain corpus and we attack an aspect of this twofold problem: the assignment of the correct relational concept given a prototypical linguistic realisation. That is, given a prototypical form "*Company has a positive net income of Currency in TimePeriod*" find *have-revenues* as the correct semantic frame. We leave apart the problem of mapping arguments to thematic roles.

We here explore the extent and nature of the general semantic knowledge truly required for the task, and, consequently, the usability of general-purpose resources such as WordNet [11]. We propose to use well assessed machine learning algorithms based on the feature vector model to study this problem. Limits of the feature vector model when applied to natural language processing tasks are discussed (Sec. 3.1). Trying to overcome these limits we propose an original model, the *verb semantic prints*, for exploiting ambiguous semantic information within the feature vector model (Sec. 3.3). In order to understand the effectiveness of the overall model we study it contrastively with a baseline model based on lexicalised syntactic information (Sec. 3.2). We argue that if general semantic information is relevant we should be able to demonstrate that the related space outperforms the other should across different machine learning algorithms. Moreover, it should demonstrate to better converge to the final classification in unsupervised clustering methods. The experimental investigation is described in Sec. 4. Results over a large range of different machine learning algorithms (collected in [17]) are compared. Finally, before concluding, we briefly discuss the related approaches (Sec. 5) as the problem of finding equivalent linguistic forms for relational concepts is largely debated.

2 A "terminological" perspective in learning equivalent linguistic forms

Domain corpora naturally contain a large quantity of domain knowledge: the same knowledge needed for adapting or building semantic models for semantic-oriented applications. A common practice in terminology extraction [8] is to exploit this knowledge trying to study what emerges from the textual collections. The problem there is to build terminological dictionaries containing relevant concepts, i.e. *terms*.

Our target is to examine domain corpora in order to find relevant *relational concepts* (i.e. semantic frames) and their corresponding linguistic realisations. In analogy with termi-

nological studies, we define a notion of *admissible surface form* (i.e. prototypes for possible textual representations) for relational concepts. Generally prototypes are given at the syntactic level. We expect that the linguistic forms of *relevant* relational concepts regularly emerge from a possibly complex (but domain independent) corpus analysis process. This can help in both deciding the relevant relational concepts and finding the one-to-many mappings with the linguistic realisations. In this process the following steps are undertaken:

1. *Corpus processing*: the *admissible surface forms* are detected and syntactically normalised. Each normalised form is a generalization of several observations ranked according to their *domain relevance* (i.e. their frequency).
2. *Concept formation*: the most important normalised forms are selected and they provide the set of target conceptual relationships. We will refer to this set as *T*.
3. *Form classification*: the generalised forms are classified according to the types defined in *T*.

The notions of *admissible surface forms* and of *domain relevance* used in the corpus processing phase are borrowed from the terminology extraction practice. These are very useful in concentrating the efforts only on relevant analysed text fragments.

As we want to analyse relational concepts we will limit our attention here to verb phrases. Our admissible surface form will be a verb with all his arguments. Even if verb phrases do not cover all the possible relational words, these are very good indicators. We will assume that the concepts, i.e. the catalogue of the named entities and the terms, are given.

The first phase is done more or less automatically using the technique introduced by [5]. Then, domain experts, exposed to the data such as the ones in Tab. 1 sorted according to their relevance (e.g. computed on the frequency *freq* of the form), can define the relational concepts. In the example *Cooperation/Splitting among Companies* (2-1) or *Market trends* (6-1) can be the two relational concepts formed in this phase. Finally, the classification of the instances is done accordingly, i.e. the column *relational concept* is compiled. The *concept formation* phase is naturally more difficult than the actual classification even if in this phase the concepts as 6-1 and 2-1 are defined extensionally. In the *classification phase* experts using the surrogate forms are able to decide the concept extension on the basis of the observable features such as *percent_ne* (*percentage*), *entity_ne* (*named entity*), *share*, *fall*, *lose*, *join*, or *own*.

<i>freq</i>	<i>generalised form</i>	<i>relational concept</i>
88	(subj,entity_ne) own (dirobj,percent_ne)	1-2
70	(subj,entity_ne) join (dirobj,entity_ne)	1-2
58	(subj,entity_ne) lose (dirobj,percent_ne)	6-1
47	(subj,share) fall (dirobj,percent_ne)	6-1

Table 1. A very small sample of the classified admissible forms

3 Syntactic feature space and verb semantic prints for learning relational concepts

The purpose of this study is trying to imitate experts in forming relational concepts and in classifying linguistic forms using well-assessed machine learning algorithms. We want also to investigate the role in the task of general semantic knowledge (i.e. Wordnet). Before developing new algorithms for a task it can be useful to understand if the feature observation space is worthy. However, the basic problem that arises when using existing machine learning algorithms is to understand if the underlying model, i.e. the feature-value vector and its usage, supports the observations we want to model. Before describing the syntactic (Sec. 3.2) and the semantic model (Sec. 3.3) we propose for form classification and, eventually, for relational concept formation, we examine the limitations of the feature-value vector model when used over models for natural language (Sec. 3.1).

3.1 Feature-Value Vector vs. Syntax and Concept Hierarchies

A largely used model for describing instance characteristics is the feature-value vector. This model underlies many machine learning algorithms as the ones gathered in [17]. It suggests an observation space in which dimensions represent features of the object we want to classify and dimension values are the values of the features as observed in the object. Each instance object is then a point in the feature space, i.e. if the feature space is (F_1, \dots, F_n) an instance I is:

$$I = (f_1, \dots, f_n) \quad (1)$$

where each f_i is respectively the value of the feature F_i for I .

Many machine learning algorithms (as the ones in [17]) use the feature-value model assuming:

- the *a-priori independence*: each feature is *a priori* independent from the others and, therefore, no possibility is foreseen to make explicit relations among the features;

- the *flatness* of the set of the values for the features: no hierarchy among the values of the set is taken in consideration;
- the *certainty of the observations*: given an instance I in the feature-value space, only one value is admitted for each feature.

Under these limitations they offer the possibility of selecting the most relevant features that may decide whether or not an incoming object in the feature-value space is instance of a given concept.

Exploiting the feature-value vector model and the related learning algorithms in the context processing natural language may then be a very cumbersome problem especially when the successful bag-of-words abstraction [15] is abandoned for deeper language interpretation models. The *a-priori* independence among features, the flatness of the values, and the certainty of the observations are not very well suited for syntactical and semantic models. On the one side, syntactical models would require the possibility of defining relations among features in order to represent either constituents or dependencies among words. On the other side, a semantic interpretation of the words (intended as their mapping in an is-a hierarchy such as WordNet [11]) would require the possibility of managing hierarchical value sets in which the substitution of a more specific node with a more general one can be undertaken as generalisation step. Finally, the ambiguity of the interpretations (either genuine or induced by the interpretation model) stresses the basic assumption of the *certainty of the observations*. Due to ambiguity, a given instance of a concept may be seen in the syntactic or the semantic space as set of alternative observations. The limits of the underlying interpretation models in selecting the best interpretation requires specific solutions to model *uncertainty* when trying to use feature-value-based machine learning algorithms for learning concepts represented by natural language expressions.

3.2 A very simple syntactic (lexicalised) model

As we have seen in Sec. 2, the objects to be classified are generalised verb forms, i.e. verbs with their more frequent arguments. Apparently, it can seem very simple mapping those structures to the feature-value vector. The

verb and the more stable arguments are in fact highlighted and, moreover, the arguments are classified according to the played syntactic role. A straightforward mapping can therefore be performed and this is what we did. We call this space as the syntactic-lexicalised feature space, hereafter referred as *synt-lex* space. The selected features (for the feature-value vector model) are then respectively the verb, the subject, the object, and finally the remaining arguments represented by their heading preposition. This defines the feature vector (F_1, \dots, F_n) . Each pattern prototype $(v, \{(arg_1, lex_1), \dots, (arg_n, lex_n)\})$ has therefore a mapping to a feature-value vector in the following way. Each F_i has the value:

$$f_i = \begin{cases} v & \text{if } F_i = verb \\ lex_j & \text{if } \exists j. F_i = arg_j \\ none & \text{otherwise} \end{cases} \quad (2)$$

It is worth noticing that in the case of the prototype forms the syntactic ambiguity is not a problem. These patterns are in fact abstractions of the behaviour of the verbs in the corpus, i.e. its arguments are statistically filtered. Furthermore, in the corpus processing phase, first of all stable generalised noun phrases are detected. This helps to filter out possibly frequent wrong verb attachments detected by the syntactic parser. Therefore, each item in the verb prototype form is then unambiguously considered as verb argument.

The chosen mapping method has some inherent limitations. Firstly, the structure of the complex noun phrases is not resolved in the feature-value model. They are in fact preserved as they are, i.e. the overall structure is replicated in the value of the related feature. For instance, in the case of *(subject, share_of_companyNE)* where a complex noun phrase appear the value given to the *subject* feature is exactly the related form. The main reason for this choice is that the more complex structure is more selective for classifying incoming instances. However, no subsumption is possible between the form *share_of_companyNE* and *share*. Such instances will be considered as completely different forms. The second limitation is instead introduced by the "variable drop" we perform in building the verb pattern prototypes. As part of the semantic of the verb is given by its surface syntactic structure [10], we tend also to offer relevant partially incomplete verb pattern prototypes where the lexicalisation of some syntactic argument may be left ungrounded. The annotators may face a pattern as the following:

(fall, \{(subject, ANY), (from, currencyNE), (to, ANY)\})

where some arguments of the verb are indicated but no restriction is given (i.e. ANY lexicalisation or named entity class is admitted). Possibly using the expectations induced by the investigated domain, the annotator should decide

whether or not the given information helps in classifying the instance. In some case, a decision may be also taken with this reduced information. However, as no subsumption is possible in the feature-value translation of the instance no explicit relation may be drawn with an other instance such as *(fall, \{(subject, share)\})*. This sort of variable drop cannot be managed. Finally, the mapping solution we adopted does not take into account the possible syntactic changing of the arguments as considered in the method exploited in [9] for a verb paraphrasing algorithm. It is worth noticing that in the case of [9] the search space was reduced by the fact that only couples of verbs suggested by a dictionary have been considered.

3.3 Ambiguous conceptual generalisations as verb fingerprints

The exploitation of conceptual hierarchy is instead a more cumbersome problem due to the limits of the feature-value model. The idea here is to investigate the possibility of integrating some sort of "semantic" generalisation for the verbs. These latter semantically govern the verbal phrases taken as forms admissible for the relationships and may give an important input to cluster prototype forms in classes. For instance, let us take the patterns in Tab. 1 and suppose that the first three lines have already been encountered, i.e. these can be considered training examples. According to the syntactic-lexicalised space previously defined the new instance may belong both to class 6-1 and to the class 1-2 as it has one common feature with all the considered known instances. The only possibility of classifying the new instance in one of the two classes relies on some sort of generalisation and the verb seems to be a very good candidate. According to WordNet *lose* and *fall* have two common ancestors *change* and *move-displace*. This does not happen for *fall* and *join* or *fall* and *own*. The injection of such a kind of knowledge seems therefore to be useful for the classification task as happens in [4, 9] where noun conceptual hierarchies have been exploited using the definition of distance measures among nodes.

The introduction of a conceptual hierarchy is somehow in contrast with what has been above called the *flatness* of the feature values. If we want to use this information, this hierarchies should be somehow reduced to a flat set where the problem of the inherent structure is simply forgot. One possibility is choosing one level of generalisation and reducing each element to this level. This is the one we adopt in our model for the exploitation of conceptual hierarchies in the problem of detecting equivalent surface forms. In particular, in order to limit the number of features we have chosen the level of the topmosts, hereafter referred as the set T .

If the previous choice helps in using part of the hierar-

chy, there is still the issue of the ambiguity that in this case cannot be neglected. We do not plan to use any a priori word sense disambiguation mechanism. We would rather prefer to discover and limit the senses of the investigated verb a posteriori, i.e. while analysing the verb prototype forms. Verb senses should be determined in the domain defined by the text collection. The ambiguity should then be modelled in the images of the pattern prototypes in the feature space. It is as if we model uncertainty in the observations of concept instances. However, features can not have multiple values. The way we propose in our model to solve the problem is to use all the topmost senses activated by the analysed verb as representing of the "overall sense" of the verb. This set can be considered as *verb semantic print*. It will be the task of the machine learning algorithm the selection of the sense (or the senses) more promising for representing the investigated relationship. The algorithm will therefore also work as verb sense disambiguator if the semantic information and the way we use it demonstrates to be useful.

The second model we propose integrates then syntactic with semantic information. The syntactic semantic space is $(F_1, \dots, F_n, T_1, \dots, T_k)$ where F_i features and the related f_i values have been defined in the previous section whilst the T_j represent the *verb semantic print*. In particular, all the elements in the topmost set T are represented in the feature space. Given a verb prototype form headed by the verb v , the value $t_i \in \{yes, no\}$ for the each semantic feature T_i in the respective point in the feature space is obtained as follows:

$$t_i = \begin{cases} yes & \text{if } hyper(v, T_i) \\ no & \text{otherwise} \end{cases} \quad (3)$$

where $hyper(x, y)$ is the property defining the hyperonym relation among x and y . This latter space will hereafter referred as syntactic-lexicalised-semantic space(*synt-lex-sem*).

4 Experimental investigation

In the previous sections, we proposed a model for exploiting syntax information and semantic networks in machine learning algorithms. As discussed, the proposed models (and the related feature spaces) relies on a large number of approximations to overcome the limitations of the feature-value model. In this section, we will explore the performances the machine learning algorithms will obtain relying on the proposed models in order to understand the relevance of the syntactic and semantic information. First of all, we will describe the test set preparation. This will clarify the final classification task. Secondly, the performances of a number of machine learning algorithms will be analysed over the two proposed feature-value space, i.e. *synt-lex* and *synt-lex-sem*. In this latter phase we will use well-assessed machine learning algorithms gathered in Weka [17]. This

collection of algorithms, originally done for Data Mining, has the principal advantage of proposing stable input interfaces for a large number of algorithms. This speeds up the possibility of testing a large number of different algorithms for the same problem. The cross-algorithm validation can give hints on the relevance of the chosen features and on the correctness of the proposed model.

4.1 Corpus analysis and test-set preparation

As discussed in Sec. 2, the context of the experiment is an overall methodology intended to extract equivalent forms out from a homogeneous document collection, i.e. the domain corpora. It worth noticing that the homogeneity hypothesis seems to be similar to the one driving the methods in [18, 14]. The main difference is the grain: the cited two methods in fact that it is stated for each document the belonging to a very specific class representing the specific information need, conservatively here we are thinking to documents related to a coarse grain class such as *sport*, *finance*, etc. Efficient methods to obtain such a document classification may be settled on the bag-of-words document model [15]. Moreover, such kinds of classified corpora are largely available: news agencies and on-line newspapers tend to offer documents organised in a classification scheme to better serve their costumers.

For the reported experiment, we used a corpus consisting of financial news. The text collection gathers around 12,000 news items published from the Financial Times in the period Oct./Dec. 2000. The relational concepts we will discover are therefore the ones related to financial events. After the *corpus processing phase*, that selected around 44,000 forms appearing more that 5 times in the corpus collection, in the *concept formation phase* 13 target relational concepts have been defined inspecting the top ranked forms (see Tab. 2). Even if we don't claim this as an exhaustive list, the defined relational concepts represent the more relevant knowledge appearing in the document collection and, more in general, in financial news.

The classification of the forms in the classes has been performed by 2 human experts. Out of the first 2,000 forms considered, 497 were retained as useful, i.e. the information carried in the words or in the named entity classes survived in the form has been considered sufficient to draw a conclusion on the classification. Due to the nature of the overall list of pattern prototypes, some of the more specific forms may be trivially tagged using an eventually classified more general form. In the preparation of the final test set we therefore got rid of this simple cases. When the class of the more specific form it is the same of the more general one, the more specific form has been removed. The resulting test set consists then of 167 different forms whose classification cannot be trivially obtained. The distribution of the forms

	Class	# of equivalent linguistic forms
1	RELATIONSHIPS AMONGS COMPANIES	
	1-1 Acquisition/Selling	15
	1-2 Cooperation/Splitting	8
2	INDUSTRIAL ACTIVITIES	
	2-1 Funding/Capital	4
	2-2 Company Assets (Financial Performances , Balances, Sheet Analysis)	20
	2-3 Market Strategies and plans	
	2-4 Staff Movement (e.g. Management Succession)	6
	2-5 External Communications	13
3	GOVERNMENT ACTIVITIES	3
	3-1 Tax Reduction/Increase	
	3-2 Anti-Trust Control	
4	JOB MARKET - MASS EMPLOYMENT/UNEMPLOYMENT	3
5	COMPANY POSITIONING	
	5-1 Position vs Competitors	3
	5-2 Market Sector	7
	5-3 Market Strategies and plans	7
6	STOCK MARKET	
	6-1 Share Trends	62
	6-2 Currency Trends	0

Table 2. The event class hierarchy of the financial domain and form distribution

in the classes is reported in Tab. 2.

It is worth noticing that in the final list only 4 macroscopic parsing errors survive: 3 related to prepositional phrase headed by *of* erroneously considered attached to the verb and one related to the form:

$(value, \{(diobj, company_at_currencyNE)\})$

The verb modifier *at_currencyNE* has been erroneously considered as modifier of the noun *company*. This is mainly because the overall form appears frequently as it is and, therefore, the fact that is chosen the "noun" reading is because this attaching phase is run as the first. These errors have been left in the final list in order to see the robustness of the learning algorithms with respect to spurious input data.

4.2 Analysis of the results

The classification problem over the proposed spaces has been therefore studied with a number algorithms and the results have been reported in tab. 3. It appears that the base-line of the classification problem proposed is around 37% that is reached by those algorithms classifying all the instances in the more probable class (i.e. 6-1). This value of performance is obtained by the NaiveBayes classifier and the DecisionStump. An important observation is that all the other algorithms report even in the syntactic space better results with respect to the base-line, i.e. they are not confused by the provided features. Furthermore, the use of the semantic information by means of the *verb semantic print* seems to be relevant. The major part of the investigated algorithms has an advantage in semantic space. The confusion introduced by the ambiguity seems to be easily managed and the relevant information used. The algorithms are doing the job of disambiguating the verb senses. The best result is obtained by the Voting Feature Interval algorithm on the semantic space. However, it does not seems to have

a relevant improvement with the introduction of the semantic. It is worth noticing that this model is statistically based and, when it faces nominal attributes¹ as the one proposed here, it becomes very similar to a profiled based classifier. Looking in the tab. 3, it furthermore seems that algorithms classifying with probability scores (as the NaiveBaye, HyperPipes, and VFI) take a small benefice from using the semantic information as it has been modelled.

Algorithms based on the decision trees (i.e. j48) give moreover the possibility to understand which are the more important attributes driving the decisions. Observing the decision tree for the *synt-lex-sem* space, it becomes clear that the more selective information is represented by the verb senses. Verb lemmas nearly disappeared, i.e. verb senses generalised this information. This phenomenon is not obvious due to the previous independence among the attributes. Furthermore, interesting classification rules as the followings may be observed:

$$\left(\begin{array}{c} \text{verb} \\ \text{change} \\ \neg \text{control} \end{array} \right) \left\{ \begin{array}{l} \left(\begin{array}{c} \text{obj} \\ \text{percentNE} \\ \text{currencyNE} \end{array} \right) \quad \mathbf{6-1} \\ \left(\begin{array}{c} \text{obj} \\ \text{capital} \\ \text{fund} \end{array} \right) \quad \mathbf{2-1} \\ \left(\begin{array}{c} \text{subj} \\ \text{turnover} \\ \text{income} \\ \text{operating-profit} \\ \text{pre-tax-profit} \\ \text{revenue} \\ \text{profit} \end{array} \right) \quad \mathbf{2-2} \end{array} \right. \quad (4)$$

A verb of *change* (but having any sense of *control*) assumes very different meaning according to the companions. This clustering can be a very interesting starting point to write more complex semantic restrictions that tend to cluster also

¹ Attributes assuming values in a finite set.

<i>Method</i>	<i>synt-lex</i>	<i>synt-lex-sem</i>	% increase/decrease
j48.J48	60.355%	65.0888%	+7,84%
j48.PART	53.8462%	56.8047%	+5,49%
DecisionStump	36.6864%	42.0118%	+14,52%
DecisionTable	59.1716%	59.1716%	0
IB1	47.3373%	60.9467%	+28,75%
IBk	55.6213%	60.9467%	+9,57%
ID3	44.9704%	44.9704%	0
NaiveBayes	36.6864%	37.2781%	+1,61%
HyperPipes	63.9053%	62.7219%	-1,85%
VFI	65.6805%	66.2722%	+0,90%

Table 3. Success rate of different methods over the two spaces in a 5-fold cross-validation

nouns as done in [4, 9].

There is a last consideration in favour of the semantic space. It seems to offer a better possibility of learning this classes from scratch using a clustering algorithm. In the case a very simple algorithm, i.e. the simple k-means with 20 clusters and averaged on 10 different seeds we obtained an error rate of 72.54% for the synt-lex space and 69.17% for the synt-lex-sem space. This timid result induces to think that, in the concept formation phase, better results can be obtained using some sort of semantic model.

5 Related work

It is largely agreed that availability of explicit many-to-one mappings between linguistic forms and their corresponding meaning (i.e. concepts or relational concepts) is beneficial for several linguistic applications. Many researches are in fact devoted to propose methods for automatically building equivalence classes of patterns in fields such as Information Extraction [18, 14], Question Answering [13], Terminology Structuring [12], or Paraphrasing [3, 9].

The automatic construction of equivalent linguistic patterns has been studied attacked from extremely different perspectives and for apparently different reasons. The target relationships range from the very general *hyperonym* relation investigated in automatic approaches to terminology structuring (e.g. [12]) to more specific information as those expressed by equivalence classes of paraphrases [3, 8, 9]. Clearly, template acquisition as typically employed in Information Extraction (e.g. [14]) is part of these studies. The target relationships may vary slightly but the common underlying targets of these methods are equivalence relations derived by analysing text material. The aim is to derive different surface forms of prototypical relationships by means of the smallest annotation effort possible.

In [14, 18] the problem of building information extraction patterns from scarcely annotated texts is investigated.

In this case, the target relationship is very complex (i.e. a template) and very specific. Due to the fact that the template is *a priori* known, the notion of *relevance* of the texts in its respect can be suitably exploited. Similarities among the different but *relevant* texts suggest equivalent linguistic forms. The issue of classifying texts is central in the two approaches: in [14] the full classification of the texts in relevant vs. irrelevant is required whilst in [18] a bootstrapping approach is used². It is to be noticed that both methods strongly rely on the shortness of the investigated texts, each one usually targeted to only one template.

A completely different approach to pattern clustering is proposed in [12, 13]. The targets are binary relationships among concepts and the assumption is that (at least some instances of) the related concepts are known *a priori*. When such coupled concepts jointly appear in a text fragment, this latter is assumed as a valid form for the target relationship.

In [12], the method has been used to compile equivalent forms for the *is-a* relationship in the context of terminology structuring. As in any terminology extraction approach the corpus used specifically models a knowledge domain. In [13], the corpus considered has been the entire world wide web and the target was to find the answering patterns using question-answer couples. Questions are first of all (manually) clustered to identify the target relationship types, called here question types (e.g. *inventor*, *discoverer*, etc.). Then for each question the couple *answer* and main *name* of the question are extracted. These latter are used to query an information retrieval engine in order to find the forms representing the given relationships.

In [3] the target is to learn syntactic paraphrasing rules mainly for verbal sentences instead of nominal forms (e.g. as in [8]). The problem is then slightly different but an interesting method for deriving the equivalence among the surface forms is used. In fact, "parallel corpora", as those employed in machine translation studies, are collected by

²The relevance of texts with respect to a template is modelled as a sort of distance between new texts and a kernel of annotated texts

groupings different English translations of a single non-English text (e.g. a novel). The different translator styles offer heterogeneous translations of the same sentences that in fact convey the same meaning. Parallel sentences thus embody equivalent forms of the same relationship. Although this method is very interesting for general syntactic paraphrasing rules, it has a limited applicability due to the specific "parallel corpora" employed.

For all the methods, the use of some previous specific knowledge (not always available) seems indispensable:

- focused and structured templates plus examples in [18, 14]
- definitions and examples of the target relationships in [12, 13]
- parallel corpora for [3]

6 Conclusions

In this paper, after the analysis of the limits of the feature-value model, we proposed a method for exploiting well-assessed machine learning algorithm for the problem of learning equivalent surface forms. We obtained some indications that the proposed way to use semantic hierarchies may be helpful in the proposed problem. In any case, the overall approach may be included as a suggesting mechanism for the experts involved in the task.

References

- [1] D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson. Fastus: a finite-state processor for information extraction from real-world text. In *13th International Joint Conference on Artificial Intelligence*, Chambry, France, 1993.
- [2] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The Berkeley framenet project. In *Proceedings of the COLING-ACL*, Montreal, Canada, 1998.
- [3] R. Barzilay and K. McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th ACL Meeting*, Toulouse, France, 2001.
- [4] R. Basili, M. T. Pazienza, and M. Vindigni. Corpus-driven learning of event recognition rules. In *Proceedings of Workshop on Machine Learning for Information Extraction, held in conjunction with the 14th European Conference on Artificial Intelligence (ECAI)*, Berlin, Germany, 2000.
- [5] R. Basili, M. T. Pazienza, and F. M. Zanzotto. Learning IE patterns: a terminology extraction perspective. In *Proc. of the Workshop of Event Modelling for Multilingual Document Linking at LREC 2002*, Canary Islands (Spain), 2002.
- [6] R. Gaizauskas and K. Humphreys. Using a semantic network for information extraction. *Natural Language Engineering*, 3, Parts 2 & 3:147–169, 1997.
- [7] D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, Nov. 2002.
- [8] C. Jacquemin. *Spotting and Discovering Terms through Natural Language Processing*. Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 2001.
- [9] N. Kaji, D. Kawahara, S. Kurohashi, and S. Sato. Verb paraphrase based on case frame alignment. In *Proceedings of the 40th ACL Meeting*, Philadelphia, Pennsylvania, 2002.
- [10] B. Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, IL, 1993.
- [11] G. A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, Nov. 1995.
- [12] E. Morin. *Extraction de liens sémantiques entre termes à partir de corpus de textes techniques*. PhD thesis, Université de Nantes, Faculté des Sciences et de Techniques, 1999.
- [13] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th ACL Meeting*, Philadelphia, Pennsylvania, 2002.
- [14] E. Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, Portland, Oregon, 1996.
- [15] G. Salton. *Automatic text processing: the transformation, analysis and retrieval of information by computer*. Addison-Wesley, 1989.
- [16] D. Stallard. Talk'n'travel: A conversational system for air travel planning. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP'00)*, Seattle, Washington, 2000.
- [17] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, Chicago, IL, 1999.
- [18] R. Yangarber. *Scenario Customization for Information Extraction*. PhD thesis, Courant Institute of Mathematical Sciences, New York University, 2001.

Automatic Acquisition of Taxonomies from Text: FCA meets NLP

Philipp Cimiano, Steffen Staab and Julien Tane

Institute AIFB

University of Karlsruhe

{cimiano,staab,tane}@aifb.uni-karlsruhe.de

Abstract

We present a novel approach to the automatic acquisition of taxonomies or concept hierarchies from domain-specific texts based on Formal Concept Analysis (FCA). Our approach is based on the assumption that verbs pose more or less strong selectional restrictions on their arguments. The conceptual hierarchy is then built on the basis of the inclusion relations between the extensions of the selectional restrictions of all the verbs, while the verbs themselves provide intensional descriptions for each concept. We formalize this idea in terms of FCA and show how our approach can be used to acquire a concept hierarchy for the tourism domain out of texts. We then evaluate our method by considering an already existing ontology for this domain.

1 Introduction

Taxonomies or conceptual hierarchies are crucial for any knowledge-based system, i.e. any system making use of declarative knowledge about the domain it deals with. Within natural language processing (NLP), information extraction or retrieval systems for example can profit from a taxonomy to provide information at different levels of detail. Machine-learning based IE systems such as described in (Ciravegna, 2001) could for example identify concepts at different levels of abstraction with regard to

a given concept hierarchy. Furthermore, the integration of a concept hierarchy would also enable such systems to generalize semantically over words and thus produce more compact and concise extraction rules. In information retrieval (IR), the availability of an ontology for a certain domain allows to replace traditional keyword-based approaches by more sophisticated ontology-based search mechanisms such as the one proposed in (Guarino et al., 1999). In general it is clear that any form of syntax-semantics interface will become more transparent and concise if a conceptual hierarchy is available.

However, it is also well known that every knowledge-based system suffers from the so called *knowledge acquisition bottleneck*, i.e. the difficulty to actually model the knowledge relevant for the domain in question. In particular ontology development is known to be a hard and time-consuming task.

In this paper we present a novel method to automatically acquire taxonomies from domain-specific texts based on Formal Concept Analysis (FCA), a method mainly used for the analysis of data (Ganter and Wille, 1999). The main benefits of our method are its adaptivity as it can be applied to arbitrary corpora and domains, its speed (compared to the process of hand-coding an ontology) as well as its robustness in the sense that it will not fail due to social aspects as present in traditional ontology development projects. Furthermore, if the corpora are updated regularly, it is also possible to let the ontology evolve according to the changes in the corpus. This is in line with the corpus and domain-specific form of lexicon as envisioned in (Buitelaar, 2000).

2 The Underlying Idea

An ontology is a formal specification of a conceptualization (Gruber, 1993). A conceptualization can be understood as an abstract representation of the world or domain we want to model for a certain purpose. The ontological model underlying this work is based on the one in (Bozsak et al., 2002):

Definition 1 (Ontology)

An ontology is a structure $O := (C, \leq_C, R, \sigma, \leq_R)$ consisting of (i) two disjoint sets C and R called *concept identifiers* and *relation identifiers* respectively, (ii) a partial order \leq_C on C called *concept hierarchy* or *taxonomy*, (iii) a function $\sigma : R \rightarrow C^+$ called *signature* and (iv) a partial order \leq_R on R called *relation hierarchy*, where $r_1 \leq_R r_2$ implies $|\sigma(r_1)| = |\sigma(r_2)|$ and $\pi_i(\sigma(r_1)) \leq_C \pi_i(\sigma(r_2))$ for each $1 \leq i \leq |\sigma(r_1)|$.

Furthermore, for each ontology O we will define a lexicon L_O as well as a mapping $F_O : C \rightarrow 2^{L_O}$ by which each concept is mapped to its possible lexical realizations. In addition, we will also consider the inverse function $F_O^{-1} : L_O \rightarrow 2^C$. Thus in our model a concept can be expressed through different expressions (*synonyms*) and one expression can refer to different concepts, i.e. expressions can be *polysemous*.

The aim of the approach presented in this paper is now to automatically acquire the partial order \leq_C between a given set of concepts C . The general idea underlying our approach can be best illustrated with an example. In the context of the tourism domain, we all have for example the knowledge that things like a *hotel*, a *car*, a *bike*, a *trip* or an *excursion* can be booked. Furthermore, we know that we can rent a *car*, a *bike* or an *apartment* and that we can drive a *car* or a *bike*, but only ride a *bike*. Moreover, we know that we can join an *excursion* or a *trip*. We can now represent this knowledge in form of a matrix as depicted in table 1. On the basis of this knowledge, we could intuitively build a conceptual hierarchy as depicted in figure 1. If we furthermore

	bookable	rentable	driveable	rideable	joinable
apartment	x	x			
car	x	x	x		
motor-bike	x	x	x		
excursion	x				x
trip	x				x

Table 1: Tourism domain knowledge as matrix

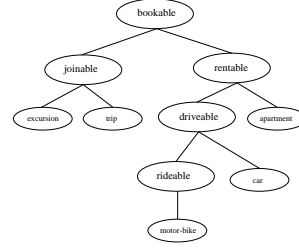


Figure 1: Hierarchy for the tourism example

reflect about the intuitive method we have used to construct this conceptual hierarchy, we would come to the conclusion that we have basically mapped the inclusion relations between the sets of the verbs' arguments to a partial order and furthermore have used the verbs itself to provide an intensional description of the abstract or non-lexical concepts we have created to group together certain 'lexicalized' concepts. In the next section we introduce Formal Concept Analysis and show how it can be used to formalize the intuitive method described above.

3 Formal Concept Analysis

Formal Concept Analysis (FCA) is a method mainly used for the analysis of data, i.e. for investigating and processing explicitly given information. Such data are structured into units which are formal abstractions of concepts¹ of human thought allowing meaningful comprehensible interpretation (Ganter and Wille, 1999). Central to FCA is the notion of a *formal context*:

Definition 2 (Formal Context)

A triple (G, M, I) is called a **formal context** if G and M are sets and $I \subseteq G \times M$ is a binary relation between G and M . The elements in G are called **objects**, those in M **attributes** and I the **incidence** of the context.

For $A \subseteq G$ and dually for $B \subseteq M$, we define :

$$A' := \{m \in M \mid (g, m) \in I \ \forall g \in A\}$$

$$B' := \{g \in G \mid (g, m) \in I \ \forall m \in B\}$$

¹Throughout this paper we will use the notion *concept* in the sense of *formal concept* as used in FCA (see below) as well in the ontological sense as defined in section 2. The meaning should in any case be clear from the context.

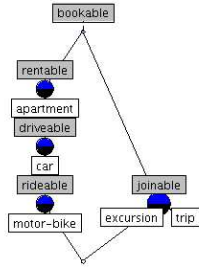


Figure 2: The tourism lattice

Intuitively speaking, A' is the set of all the attributes common to the objects in A , while B' is respectively the set of all the objects which have in common with each other the attributes in B . Furthermore, we define what a *formal concept* is:

Definition 3 (Formal Concept)

A pair (A, B) is a **formal concept** of (G, M, I) if and only if

$$A \subseteq G, B \subseteq M, A' = B \wedge A = B'$$

In other words, (A, B) is a **formal concept** if and only if the set of all attributes shared by the objects in A is identical with B and on the other hand A is also the set of all the objects which have in common with each other the attributes in B . A is then called the **extent** and B the **intent** of the concept (A, B) . The concepts of a given context are naturally ordered by the **subconcept-superconcept relation** as defined by:

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_2 \subseteq B_1)$$

Thus, formal concepts are partially ordered with regard to inclusion of their extents or (which is equivalent) to inverse inclusion of their intent.

Thus, table 1 represents the incidence I of the formal context in form of a matrix. The corresponding sub-/superconcept partial order computed by FCA is depicted in figure 2 in form of a lattice. The representation makes use of **reduced labeling** as described in (Ganter and Wille, 1999) such that each object and each attribute is entered only once in the diagram. Finally, it is just left to clarify how we obtain a concept hierarchy, i.e. our partial order

\leq_C out of a lattice such as depicted in figure 2. We accomplish this by creating for each node in the lattice a concept labeled with the intent of the node as well as a subconcept of this concept for each element in the extent of that node. Furthermore, we also remove the bottom element of the lattice and preserve the other nodes and edges. Thus, in general we yield a partial order which can be represented as a DAG. In particular, for the lattice in figure 2 we yield the partial order in figure 1.

4 FCA meets NLP

The decisive question is now where to get from the objects as well as the corresponding attributes in order to create a taxonomy by using Formal Concept Analysis. A straightforward idea is to extract verb-object dependencies from texts and turn the objects' head into FCA objects and the corresponding verbs together with the postfix 'able' into attributes.

As already mentioned before, we concentrated on texts related to the tourism domain. In particular, we used two rather small corpora. The first corpus was acquired from <http://www.all-in-all.de/english/>, a web-page containing information about the history, cultural events, accommodation facilities, etc. of *Mecklenburg Vorpommern*, a region in north-east Germany. The second corpus is a collection of texts from <http://www.lonelyplanet.com/destinations/>. The total size of both corpora together was roughly about a million words.

In order to acquire the verb-object dependencies from these corpora, we used LoPar², a trainable and statistical left-corner parser. LoPar was trained on the corpora before actually parsing them. LoPar's output was then post-processed with `tgrep` to actually yield the desired dependencies, i.e. the verbs and the heads of the objects they subcategorize. It is important to mention that with our method we are also able to get multi-word terms. Furthermore, we use a simple method to lemmatize the extracted terms and verbs by looking up the lemma of each word in the lexicon provided with LoPar.

Regarding the output of the parser, it has to be taken into account that on the one hand it can be erroneous

²[http://www.ims.uni-stuttgart.de/projekte/gramotron/ SOFTWARE/LoPar-en.html](http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/LoPar-en.html)

and on the other hand not all the verb-object dependencies produced are significant from a statistical point of view. Thus an important issue is actually to reduce the 'noise' produced by the parser before feeding the output into FCA. For this purpose, we calculate the overall probability $P(n)$ that a certain (multi-word) term n appears as direct object of a verb, the overall probability $P(v)$ for a certain verb v , the probability $P(n, v)$ that n occurs as the head of v 's object as well as the probability $P(n|v)$ that given a certain verb v , n appears in object position. Here are the corresponding formulas:

$$P(n) = \frac{f(n)}{\sum_{n' \in N} f(n')}$$

$$P(v) = \frac{f(v)}{\sum_{v' \in V} f(v')}$$

$$P(v, n) = \frac{f(v, n)}{\sum_{v' \in V} f(v')}$$

$$P(n|v) = \frac{f(n, v)}{f(v)}$$

where N and V are respectively the set of all terms appearing as direct objects of a verb and the set of all verbs with a direct object, $f(n)$ and $f(v)$ are respectively the number of occurrences of a term $n \in N$ as direct object and a verb $v \in V$ and $f(v, n)$ is the number of times that n occurs in the object position of v .

Now in order to weigh the significance of a certain verb-object pair (v, n) , we used three different measures: a standard measure based on the conditional probability, the mutual information measure used in (Hindle, 1990), as well as a measure based on Resnik's *selectional preference strength* of a predicate (Resnik, 1997). Here are the formulas:

Standard: $P(n|v)$

Hindle: $\log \frac{P(v, n)}{P(v)P(n)}$

Resnik: $P(n|v) * S_R(v)$

where the *selectional preference strength* of a verb is defined according to (Resnik, 1997):

$$S_R(v) = \sum_{n \in N} P(n|v) \log \frac{P(n|v)}{P(n)}$$

Thus, the selectional preference of a verb is stronger the less frequent the nouns are that appear as its direct objects. In our approach, we then only consider those verb-term pairs as attribute-object pairs for which the values of the above measures are above some threshold t . For the Formal Concept Analysis we use the *Concepts* tool downloadable from <http://www.fcacore.org.uk/>. The processing time for building the lattice was for all thresholds and measures in the worst case 12 seconds. Thus, the FCA processing time can certainly be neglected in comparison to the parsing time.

5 Evaluation

Before actually presenting the results of our evaluation, we first have to describe the task we are evaluating against. Basically, the task can be described as follows: given a set of m concepts relevant for a certain domain, order these concepts hierarchically in form of a taxonomy. Certainly, this is not a trivial task and as shown in (Maedche and Staab, 2002) human agreement on such a task has also its limits.

In order to evaluate our automatically generated taxonomies, we compare them with the tourism domain ontology developed within the GETESS project³. This ontology was primarily designed for the analysis of german texts, but also english labels are available for many of the concepts. Moreover, we manually added the english labels for those concepts whose german label has an english counterpart. As a result we yielded an ontology consisting of 1026 concepts with most of them (>95%) having an english label.

Certainly, it is not clear how two ontologies (as a whole) can be compared to each other in terms of similarity. In fact, the only work in this direction the authors are aware of is the one in (Maedche and Staab, 2002). There, ontologies are seen as a semiotic sign system and compared at a syntactic, i.e. lexical, as well as semantic level. In this line,

³http://www.getess.de/index_en.html

we present a comparison based on lexical overlap as well as taxonomic similarity between ontologies. Lexical overlap (LO) of two ontologies O_1 and O_2 will be measured as the recall of the lexicon L_{O_1} compared to the lexicon L_{O_2} , i.e.

$$LO(O_1, O_2) = \frac{|L_{O_1} \cap L_{O_2}|}{|L_{O_2}|}$$

In order to compare the taxonomy of the ontologies, we use the *semantic cotopy* (SC) presented in (Maedche and Staab, 2002). The semantic cotopy of a concept is defined as the set of all its super- and subconcepts:

$$SC(c_i, \leq_C) := \{c_j | c_i \leq_C c_j \vee c_j \leq_C c_i\},$$

where $c_j, c_i \in C$. Now, we also extend the definition to operate on sets of concepts:

$$SC(C', \leq_C) := \bigcup_{c' \in C'} SC(c', \leq_C)$$

On the basis of this definition we introduce the *common lexical cotopy* (CLC) of a lexical entry $l \in L_{O_{1,2}} = L_{O_1} \cap L_{O_2}$ with regard to O_1 and O_2 as follows:

$$CLC(l, O_1, O_2) = \bigcup_{c \in SC(F_{O_1}^{-1}(l))} F_{O_1}(c) \cap L_{O_{1,2}}$$

Taxonomic overlap \overline{TO} is now defined as follows⁴:

$$\overline{TO}(O_1, O_2) = \frac{1}{|L_{O_{1,2}}|} \sum_{l \in L_{O_{1,2}}} TO(l, O_1, O_2),$$

$$TO(l, O_1, O_2) = \frac{|CLC(l, O_1, O_2) \cap CLC(l, O_2, O_1)|}{|CLC(l, O_1, O_2) \cup CLC(l, O_2, O_1)|} 5$$

In order to compare the performance of our approach with the human performance on the task, we will interpret our results with regard to the study presented in (Maedche and Staab, 2002). In this study, five subjects were asked to model a taxonomy on the basis of 310 lexical entries relevant for the tourism domain. The taxonomic overlap (\overline{TO}) between the manually engineered ontologies reached from 47% to 87% with an average of 56.35%. Thus, it is clear that any automatic approach to derive a conceptual hierarchy between a set of concepts has definitely its limits.

⁴Here we assume that $|L_{O_{1,2}}| > 0$; otherwise \overline{TO} will be 0.

⁵The reader is referred to (Maedche and Staab, 2002) for some concrete examples for these measures.

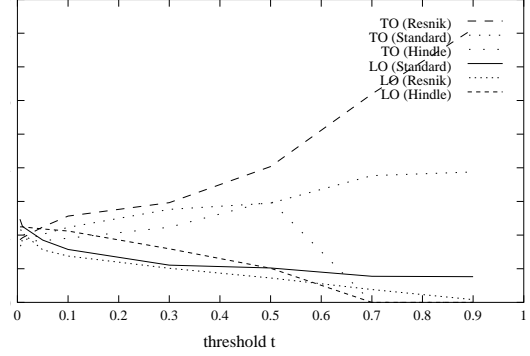


Figure 3: LO and \overline{TO} values over threshold t

5.1 Results

We generated different taxonomies with the approach described in section 4 by using the measures *Standard*, *Resnik* and *Hindle* as well as different values for the threshold t . In particular, we used the values $t \in \{0.005, 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9\}$. The size of these ontologies ranged from 0 to 7874 concepts. We then compared these 24 (=3x8) automatically generated taxonomies against the taxonomy of the GETESS ontology in terms of *lexical overlap* (LO) and *taxonomic overlap* \overline{TO} . The results of this comparison are depicted in figure 3. The best results in terms of \overline{TO} are certainly achieved by the *Resnik* measure having always the highest \overline{TO} with a small LO between 22.02% ($t = 0.005$) and 0.8% ($t = 0.9$). Interestingly, the *Standard* measure shows a worse \overline{TO} than the *Resnik* measure, but a more stable LO between 24.71% ($t = 0.005$) and 7.65% ($t = 0.9$). The performance of the *Hindle* measure in terms of LO and \overline{TO} is certainly the worst due to the fact that LO falls down to 0 at $t = 0.7$. Though these results don't allow much conclusions to be drawn, three interesting observations can be made. First, the lexical overlap seems to be very low in general, a problem which seems definitely necessary to overcome in order to yield also higher \overline{TO} values. The second observation is that the higher the threshold t , the lower the lexical overlap and the higher the taxonomic overlap get. This is on the one hand clearly due to the fact that the higher the threshold t is, the less pairs (n, v) we feed into FCA and thus the amount of lexical categories in the taxonomy decreases. On the other hand it seems

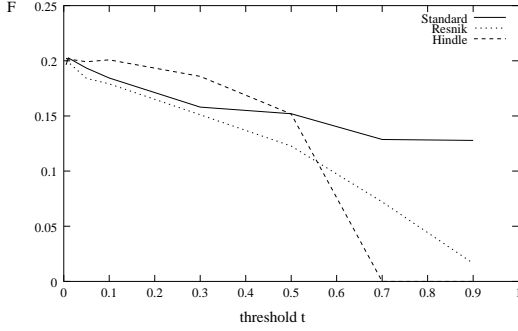


Figure 4: F-Measure (LoPar)

intuitive that it is easier to organize hierarchically fewer classes, which would explain the increasing \overline{TO} values. This leads us to the third observation, i.e. that it is important to balance \overline{TO} and LO against each other. In line with the information extraction (IE) task, in which *precision* and *recall* have to be balanced against each other, we compute also the F-Measure balancing LO and \overline{TO} , i.e. $F = \frac{2 * \overline{TO} * LO}{\overline{TO} + LO}$. The results of the F-Measures for the three measures and the different threshold values are depicted in figure 4. The three measures achieve more or less the same top results. However, the *Standard* measure definitely shows the most stable behavior. It achieves a \overline{TO} of 18.39% and a LO of 22.54% and thus a F-Measure of $F=20.25\%$ at $t = 0.01$. The best result of the *Resnik measure* ($F=20.18\%$) corresponds to a \overline{TO} of 18.62% and a LO of 22.02% at $t = 0.005$. The *Hindle* measure achieves an F-Measure of $F=20.22\%$ with a \overline{TO} of 18.40% and a LO of 22.44% at $t = 0.005$. Certainly, the major bottleneck of the approach seems to be the low lexical overlap between the ontologies.

5.2 Adding Prepositional Phrases (PPs)

The results in the previous section lead us to investigate if we could increase the lexical overlap of the automatically generated taxonomies by extracting additionally verb-PP pairs and feeding them into FCA in the same way as the verb-object pairs with the only exception that instead of adding to the verb the postfix “able” we add the postfix “_<PREPOSITION>”. The results in terms of F-Measures over the different thresholds in fact show that there was a slight increase in overall

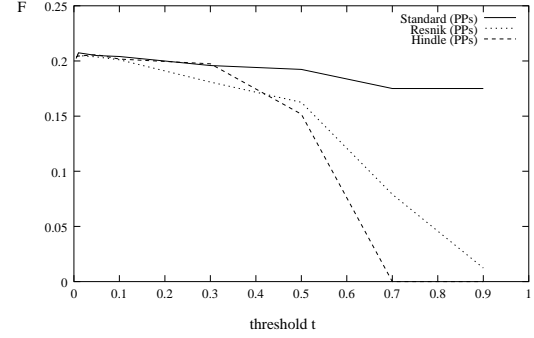


Figure 5: F-Measure (LoPar with PPs)

performance (compare figure 5). The best result ($F=20.74\%$) is again achieved by the *Standard* measure with a \overline{TO} of 16.57% and a LO of 27.71% at a threshold of $t = 0.01$. The second best result was reached by the *Resnik* measure with a \overline{TO} of 16.61% and a LO of 27.20% (and thus $F=20.63\%$) at a threshold of $t = 0.005$. The *Hindle* measure reached a F-Measure of $F=20.56\%$ corresponding to a \overline{TO} of 17.07% and a LO of 25.85% at $t = 0.05$. It seems that we have in fact increased the lexical overlap at the cost of a decrease in taxonomic overlap. However, the overall results as indicated by the F-Measures have definitely increased.

5.3 Chunking

As final experiment we substituted LoPar by Steven Abney’s chunker CASS (Abney, 1996) in the hope that due to its robustness the lexical overlap would increase even more. For this purpose we used a straightforward heuristic and interpreted every NP and PP following a verb respectively as its direct object and PP-complement. The resulting F-measures are depicted in figure 6. Interestingly, the *Standard* and *Resnik* measures seem to perform more or less the same. The best result ($F=20.21\%$) is achieved by the *Resnik* measure with a \overline{TO} of 14.08% and a LO of 34.33% at a threshold $t = 0.05$. The *Standard* measure achieves the second best result of $F=20.17\%$ at $t = 0.1$ corresponding to a \overline{TO} of 14.89% and an LO of 31.13%. The best result of the *Hindle* measure is $F=20.16\%$ with a \overline{TO} of 14.93% and an LO of 31.02% at $t = 0.5$. So the results are more or less comparable to those produced by LoPar without taking into account PPs with the difference

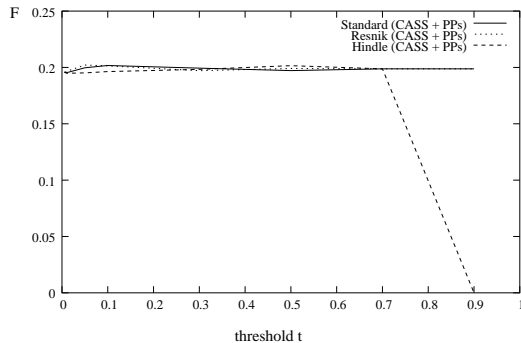


Figure 6: F-Measure (CASS)

that the lexical overlap has increased at the cost of a lower taxonomic overlap. This is certainly due to erroneous verb-object or verb-PP pairs yielded by postprocessing the chunker’s output with the above mentioned heuristic.

5.4 Discussion of Results

The experiments have shown that the best results are achieved by using a parser such as LoPar to extract verb-object and verb-PP pairs as well as the *Standard* or *Resnik* measures to select the most significant pairs. However, it has also become clear that a chunker like CASS is an attractive alternative to using a parser as it leads to only slightly worse results but is actually much faster. Overall, the best result achieved is a F-Measure of 20.74% corresponding to a \overline{TO} of 16.57% and an LO of 27.71%. In order to compare these results against human performance on the task, we first have to assess human performance in terms of the F-Measure. The assumption will be that if humans have to order hierarchically m terms they will always succeed in ordering all of them. In this sense, assuming for humans a LO of 100% and considering the average human agreement of 56.35% in terms of \overline{TO} as given above, we yield a human average performance on the task of $F=72.08\%$ which we are still quite far away from. On the other hand, when comparing the manually engineered ontologies in (Maedche and Staab, 2002) with the GETESS ontology, we get F-measure values between $F = 32.03\%$ and $F = 34.07\%$. In the light of these numbers, our results seem definitely more promising. Furthermore, the assumption of a human LO of 100% as well as the above average

agreement may hold for relatively trivial domains such as tourism, but are certainly too optimistic for more technical domains such as for example biomedicine. Thus, we believe that the more specific and technical the underlying corpus is, the closer our approach will get to human performance. In the future, we hope to support this claim with further experiments on more technical domains.

6 Discussion of Related Work

In this section, we discuss some work related to the automatic acquisition of taxonomies out of texts. The early work of (Hindle, 1990) on noun classification from predicate-argument structures is very related to the approach presented here. Hindle’s work is based on the distributional hypothesis, i.e. that nouns are similar to the extent that they share contexts. The central idea of his approach is that nouns may be grouped according to the extent to which they appear in similar verb frames. In particular, he takes into account nouns appearing as subjects and objects of verbs, but does not distinguish between them in his similarity measure. Our approach goes one step further in the sense that we do not only group nouns together, but also derive a hierarchical order between them.

Also very related to the work presented here is the approach of (Faure and Nedellec, 1998). Their work is also based on the distributional hypothesis and they present an iterative bottom-up clustering approach of nouns appearing in similar contexts. At each step, they cluster together the two most similar extents of some argument position of two verbs. However, their approach requires manual validation after each clustering step so that in our view it can not be called *unsupervised* or *automatic* anymore.

(Hahn and Schnattinger, 1998) aim at learning the correct ontological class for unknown words. For this purpose, when encountering an unknown word in a text they initially create one ‘hypothesis space’ for each concept the unknown word could actually belong to. These initial hypothesis spaces are then iteratively refined on the basis of evidence extracted from the linguistic context the unknown word appears in. In their approach, evidence is formalized in the form of quality labels attached to each hypothesis space. At the end the hypothesis

space with maximal evidence with regard to the qualification calculus used is chosen as the correct ontological concept for the word in question.

Finally, (Hearst, 1992) aims at the acquisition of hyponym relations from Grolier's American Academic Encyclopedia. In order to identify these relations, she makes use of lexico-syntactic patterns manually acquired from her corpus. Hearst's approach is characterized by a high precision in the sense that the quality of the learned relations is very high. However, her approach suffers from a very low recall which is due to the fact that the patterns are very rare.

7 Conclusion and Further Work

We have presented a method for the automatic acquisition of taxonomies out of domain-specific text which is in line with the idea of a dynamic as well as corpus- and domain-specific lexicon as presented in (Buitelaar, 2000). The method presented is certainly adaptive as it only relies on generic NLP tools. Our approach would definitely profit from some form of smoothing. We are currently experimenting with an approach to cluster the verbs into classes before actually feeding the verb-object or verb-PP pairs into FCA. Moreover, we would like to apply our approach to larger corpora, other domains as well as other languages, in particular German. In order to overcome the low lexical recall of our approach, we think that crawling texts containing the appropriate words from the WWW as presented in (Agirre et al., 2000) is a promising option. Finally, we also aim at learning other relations than taxonomic ones. For this purpose we envision an approach as described in (Resnik, 1997) in order to learn relations at the right level of abstraction with regard to our automatically acquired taxonomy. In general, we believe that a combination approach of different methodologies is the key towards automatically generating acceptable and reasonable taxonomies which can be used as a starting point for applications and then be refined during their life-cycle.

Acknowledgments Philipp Cimiano is currently supported by the IST-Dot.Kom project (<http://www.dot-kom.org>), sponsored by the EC as part of the framework V, (grant IST-2001-34038). We would like to acknowledge Martin Kavalec for kindly providing the Lonely Planet corpus. Thanks also to

the two anonymous reviewers of the ATEM Workshop for their comments.

References

- Steven Abney. 1996. Partial parsing via finite-state cascades. In *Proceedings of the ESSLLI '96 Robust Parsing Workshop*.
- E. Agirre, O. Ansa, E. Hovy, and D. Martinez. 2000. Enriching very large ontologies using the WWW. In *Proceedings of the Workshop on Ontology Construction of the ECAI*.
- E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, A. Mädche, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G. Stumme, Y. Sure, J. Tane, R. Volz, and V. Zacharias. 2002. Kaon - towards a large scale semantic web. In *Proceedings of the Third International Conference on E-Commerce and Web Technologies (EC-Web 2002)*. Springer LNCS.
- Paul Buitelaar. 2000. Semantic lexicons. In K. Simov and A. Kiryakov, editors, *Ontologies and Lexical Knowledge Bases*, pages 16–24.
- F. Ciravegna. 2001. Adaptive information extraction from text by rule induction and generalization. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*.
- D. Faure and C. Nedellec. 1998. A corpus-based conceptual clustering method for verb frames and ontology. In P. Verardi, editor, *Proceedings of the LREC Workshop on Adapting lexical and corpus resources to sublanguages and applications*.
- B. Ganter and R. Wille. 1999. *Formal Concept Analysis – Mathematical Foundations*. Springer Verlag.
- T.R. Gruber. 1993. Toward principles for the design of ontologies used for knowledge sharing. In *Formal Analysis in Conceptual Analysis and Knowledge Representation*. Kluwer.
- N. Guarino, C. Masolo, and G. Vetere. 1999. Ontoseek: Content-based access to the web. *IEEE Intelligent Systems*, 14(3):70–80.
- U. Hahn and K. Schnattinger. 1998. Ontology engineering via text understanding. In *Proceedings of the 15th World Computer Congress 'The Global Information Society on the Way to the Next Millenium' (IFIP'98)*.
- M.A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*.
- D. Hindle. 1990. Noun classification from predicate-argument structures. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 268–275.
- A. Maedche and S. Staab. 2002. Measuring similarity between ontologies. In *Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW)*. Springer.
- Philip Resnik. 1997. Selectional preference and sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*

Active Learning Selection Strategies for Information Extraction

Aidan Finn Nicholas Kushmerick

Smart Media Institute, Computer Science Department, University College Dublin, Ireland
{aidan.finn, nick}@ucd.ie

Abstract

The need for labeled documents is a key bottleneck in adaptive information extraction. One way to solve this problem is through active learning algorithms that require users to label only the most informative documents. We investigate several document selection strategies that are particularly relevant to information extraction. We show that some strategies are biased toward recall, while others are biased toward precision, but it is difficult to ensure both high recall and precision. We also show that there is plenty of scope for improved selection strategies, and investigate the relationship between the documents selected and the relative performance between two strategies.

1 Introduction

Information extraction (IE) is the process of identifying a set of pre-defined relevant items in text documents. For example, an IE system might convert free text resumes into a structured form for insertion in a relational database. Numerous machine learning (ML) algorithms have been developed that promise to eliminate the need for hand-crafted extraction rules. Instead, users are asked to annotate a set of training documents selected from a large collection of unlabeled documents. From these annotated documents, an IE learning algorithm generalizes a set of rules that can be used to extract items from unseen documents.

It is not feasible for users to annotate large numbers of documents. IE researchers have therefore investigated active learning (AL) techniques to automatically identify documents for the user to annotate [13, 12, 3].

The essence of AL is a strategy for selecting the next document to be presented to the user for annotation. The selected documents should be those that will max-

imize the future performance of the learned extraction rules. Document selection algorithms attempt to find regions of the instance space that have not yet been sampled in order to select the most informative example for human annotation. The nature of IE means that, compared to text classification, it becomes even more important to employ AL techniques. Documents are more expensive to mark-up for IE as rather than being a member of a single class, a document may contain several examples of fields to be extracted.

Several selection strategies have been studied in the more general context of machine learning. For example, confidence-based approaches select for annotation the unlabeled instance of which the learner is least confident. While such techniques are clearly applicable to IE, we focus on novel selection algorithms that exploit the fact that the training data in question is text.

AL in the context of IE is problematic, but also offers new opportunities. It is problematic in that generic approaches require feature encoding of all instances. But for LP^2 [2] and other IE systems, we need to know the details of how the learning algorithm represents a document to compute those features. This does not facilitate completely learner-independent selection strategies.

IE also offers new opportunities for AL. Because the objects in question are text, this can give rise to the possibility of using selection strategies that don't necessarily make sense in a 'generic' ML setting. For example, one of our strategies selects documents according to the frequency of common personal names.

In this paper, we investigate several selection strategies and their application to IE (Sec. 3). We show that different strategies offer a trade-off between precision or recall (Sec. 4). Some strategies improve recall at the expense of precision, while others improve precision at the expense of recall. We also estimate the optimal performance of an IE algorithm and show that there is plenty of scope for improving existing selection strate-

gies.

Furthermore, we show that the difference in performance between two selection strategies can be (weakly) predicted from the correlation between the documents they select (Sec. 5).

2 Related work

There has been a large amount of work on adaptive information extraction, e.g. [2, 1, 9] and many others. These algorithms generally perform well, but all have the potential for further improvement through active learning techniques.

Active learning refers to a variety of ways that a learning algorithm can control the training data over which it generalizes. For example, a learner might construct synthetic instances and ask the user to label them. We focus on so-called selective-sampling strategies [5], in which the learner picks an instance for the user to label from a large pool of unlabeled instances.

Selective sampling techniques are generally regarded as being of two types: confidence- or certainty-based [10], or committee-based [6]. In each case, the learner has built a model using a certain number of labeled training documents, and must select the next document to be labeled with the goal of choosing the document that will give the maximum information.

In confidence-based approaches, the learner examines unlabeled examples and attaches a confidence (usually based on the certainty with which a prediction can be made about the document) to them. Documents with low confidence are chosen to be labeled. Typically, methods for estimating certainty are based on the probability that the learner will classify a new example correctly.

In committee-based approaches, a committee of learners is constructed and each member attempts to label unlabeled documents. Documents that maximize disagreement between committee members are chosen for labeling. In fact, committee-based approaches can be regarded as confidence-based, where the confidence in a prediction is based on the agreement among committee members about that prediction.

There has been some work in the application of active learning to IE (e.g. [13, 11, 12]). [12] use learning-algorithm-specific heuristics to choose the next document for annotation. Specifically, their AL algorithm for learning Hidden Markov Models (HMM) identifies “difficult” unlabeled tokens and asks the user to la-

bel them. Difficulty is estimated by the difference between the most likely and second most likely state of the HMM.

Other applications of AL and IE do not rely on a specific learning algorithm. [13] use certainty-based sampling, where the certainty of an extracted field is the minimum of the training-set accuracies of the rules that extracted the fragment. [11] describe a multi-view approach to IE. Multi-view AL is a committee-based approach in which the committee members are formed by training on different sets of features. Muslea et al. learn two different models for extraction based on two different views of the data, and select the document where both models disagree, but are most confident in their predictions.

3 Selection strategies

3.1 Notation and terminology

The aim of an active learning selection strategy is to select documents in a way that improves performance over random selection. A selection strategy should select the document for labeling that is most informative. The difficulty is estimating how informative a document will be without knowing the labels associated with that document or the features that will represent the document. We have identified two main approaches to estimating the informativeness of a document: confidence-based and distance-based.

Confidence-based. The first approach is to try to directly estimate the informativeness of a document x using some measure of uncertainty $f(x)$. From information theory, the amount of information gained from labeling a document is equal to the uncertainty about that document before labeling it [10]. Most learning algorithms support some method of estimating confidence on unseen documents. For example, one can invoke a set of learned rules on a document, and then compute a confidence for the document based on the training-set accuracies of the rules that apply to that document. Other types of approaches such as multi-view and committee-based can also be regarded as confidence-based. Multi-view approaches estimate uncertainty using some measure of disagreement between models built using different views, while committee-based approaches estimate the confidence using agreement between committee members.

Given some confidence measure f and a pool of unlabeled documents U , a confidence-based selection strategy will pick the unlabeled document x that minimizes this measure:

$$x \equiv \arg \min_{x' \in U} f(x')$$

Distance-based. The second approach is based on the idea that for any set of instances, there is (by definition) some set of documents O that optimizes performance over the unselected documents. Furthermore, one can assume that O can be generated from some distance metric $d_O(x, x')$ over documents, by selecting the $|O|$ documents that maximize the pair-wise distance between the members of O . For example, if the learning algorithm is a covering algorithm, then performance should be maximized with a sample that covers the instance space uniformly. So the second approach is to define some distance metric $d(x, x')$ that closely approximates $d_O(x, x')$, and then sampling uniformly from that space. Rather than trying to find documents that we have low confidence in, we are trying to find documents that are different to those already seen. Specifically, given some distance metric $d(x, x')$, a set of previously selected documents S , and a pool of unlabeled data U , a distance-based selection strategy will pick the unlabeled document x that maximizes the distance from x to the members of S :

$$x \equiv \arg \max_{x' \in U} \sum_{x'' \in S} d(x', x'')$$

Of course, distance-based approaches can also be thought of as confidence-based where confidence is estimated as distance from previously seen instances. This is a less direct measure of confidence than other approaches so we feel that it warrants separate categorization.

3.2 The strategies

We introduce several novel AL document selection strategies for IE. Some of the strategies are applicable only in an IE or text classification context. While they are tailored for IE, they are generic in that they do not assume any specific IE algorithm. The learning algorithm that we use is LP^2 [2] but the active learning strategies that we investigate are not particular to our choice of learning algorithm and so we could easily substitute another IE algorithm such as BWI [9] or Rapier [1].

COMPARE. This strategy selects for annotation the document that is textually least similar to the documents that have already been annotated. We select the document that is textually most dissimilar to the documents already in the corpus. The idea is to sample uniformly from the document space, using the notion of textual similarity to approximate a uniform distribution. This is a distance-based selection strategy. Similarity can be measured in various ways, such as raw term overlap, or using TFIDF weighting but our distance metric $d(x, x')$ is the inverse of the number of words that occur in both x and x' divided by the number of words that occur in x or x' . Note that COMPARE is very fast, because the learning algorithm does not need to be invoked on the previously-selected documents in order to select the next document.

EXTRACTCOMPARE. This strategy selects for annotation the document where what is extracted from the document is textually most dissimilar to the documents in the training corpus. This is similar to Compare, except that the distance metric is $d(x, extract(x'))$, where $extract(x')$ applies the learned extraction rules to the document x' . The idea here is to select documents that don't contain text that we are already able to extract. EXTRACTCOMPARE is quite slow, because the learning algorithm must be invoked on the previously-selected documents in order to select the next document.

MELITA [4]. MELITA selects for annotation the document that matches the fewest patterns that the learning algorithm has learned from the training corpus. This is a confidence based metric. $f(x) = |extract(x)|$. This approach is similar to EXTRACTCOMPARE. It selects documents that do not match patterns that we have already learned. Like EXTRACTCOMPARE, MELITA is quite slow. Note that MELITA is essentially a special case of the approach described in [13] in that the confidences of the extracted items are ignored.

NAMEFREQ. Often the items to be extracted are people's names, but these can be difficult to extract, because they are likely to be words that the learner has not seen before. NAMEFREQ selects for annotation the document with the most unusual personal names. Specifically, NAMEFREQ assigns a part of speech tag to each term in document x , and then uses $f(x) = 1/\sum_{p \in x} n(p)$, where $p \in x$ iterates over the proper nouns in document x , and $n(p)$ is the frequency of

proper noun p as a personal name according to recent US Census data. We assume that the learner is less confident about names that are unusual as it is less likely to have seen these names before. Like COMPARE, NAMEFREQ is very fast.

BAG. Bagging is a standard approach in machine learning. We apply it to IE by invoking the learning algorithm on different partitions of the available training data and selecting the document that maximizes disagreement between the models built on different partitions of the training data. The training set is divided into two partitions and a model built using each as its training set. The document is selected where the two learners extract the most dissimilar text. This is a committee-based strategy (and thus confidence-based), where the members of the committee comprise learners built on different partitions of the training data. The confidence of prediction is estimated based on agreement between the two learned models. BAG is very slow.

ENSEMBLE. It is common in machine learning to use the combined predictions of different learning algorithms to improve performance. We can similarly with IE seek to combine selections of different selection strategies to improve learning rate. This approach is an ensemble learner based on the MELITA and NAMEFREQ strategies. It selects half of those documents that NAMEFREQ would pick and half of those that MELITA would pick. This strategy was designed after examination of the performance of the other selection strategies. The aim to try to simultaneously maximize both precision and recall. ENSEMBLE is quite slow.

4 Experiments

We have evaluated our selection algorithms on two information extraction tasks, and report our results in the form of the learning curve for each selection strategy.

Each learning curve was averaged over ten runs. Documents are added to the training-set in batches of size 10. For each selection strategy, the first 10 documents are picked at random, while subsequent batches are chosen according to the selection strategy. Each point on the learning curve shows the accuracy of the learning algorithm when trained on the selected documents and tested on the rest.

We compare our results to two baselines: a trivial strategy that selects documents randomly, and an “om-

niscient” optimal strategy. Because finding the true optimal is combinatorially prohibitive, we use a greedy estimate of the optimal (at each step, the greedy algorithm selects the one document that will result in the largest increase in performance). That is, the optimal selection x given a set of previously selected documents S and a pool U of unlabelled documents with respect to some measure M (eg, precision, recall or F1) is

$$x \equiv \arg \max_{x' \in U} M(S \cup \{x'\}).$$

We include this data as an estimate of the upper bound on the performance of any selection strategy. Finally, because even the greedy implementation requires a large amount of CPU time, we report the optimal results for just a small number of documents.

4.1 Seminar announcements

The SA dataset consists of 473 seminar announcements [7]. For each document we wish to extract the speaker, location, start-time and end-time.

Fig. 1 shows the learning curves for F1, precision and recall generated on this dataset. Looking at F1 shows that random selection is one of the better strategies. In fact only MELITA and COMPARE perform better than the random selection strategy on this extraction task, but the difference is small. However, recall that COMPARE is much faster than MELITA, so COMPARE is more suitable for the interactive scenarios that motivate MELITA [4]. NAMEFREQ performs considerably worse than the other selection strategies.

If we look at precision and recall separately, we get a clearer picture of the performance of each strategy. MELITA performs best when recall is considered followed by COMPARE and EXTRACTCOMPARE. All of these are significantly better than random. NAMEFREQ is the worst performer.

If we look at the precision learning curve, this trend is reversed. NAMEFREQ gives the highest precision, while MELITA and EXTRACTCOMPARE give the worst precision. COMPARE gives slightly better precision than random and better recall than random.

On this task, NAMEFREQ gives the best improvement in precision, while it is the worst when recall is considered. Conversely MELITA offers the best improvement in recall, but performs worst when precision is considered.

Each strategy seems to bias toward either improving precision or improving recall. Some strategies can be

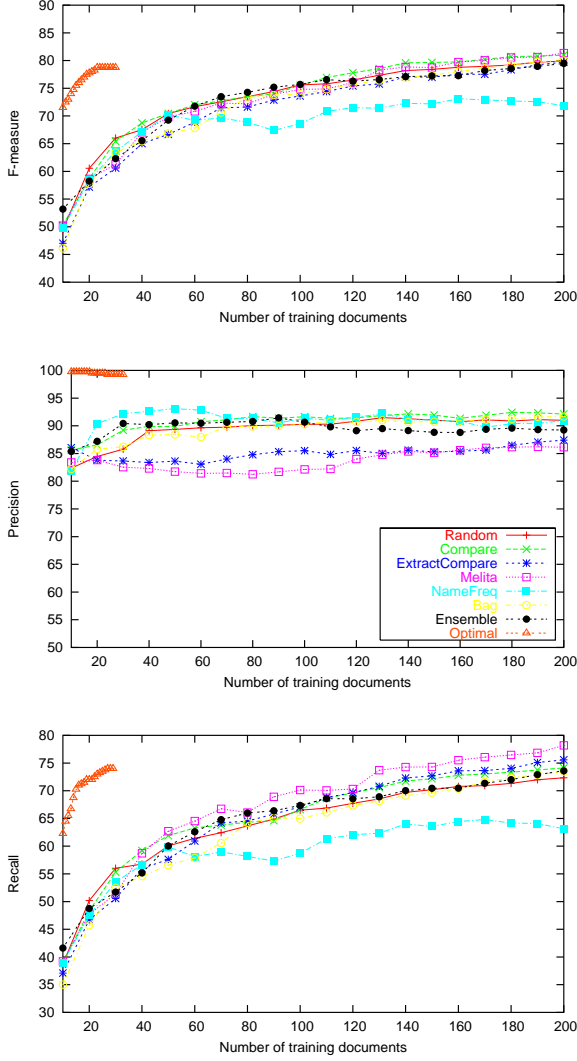


Figure 1: Learning curves for the SA dataset.

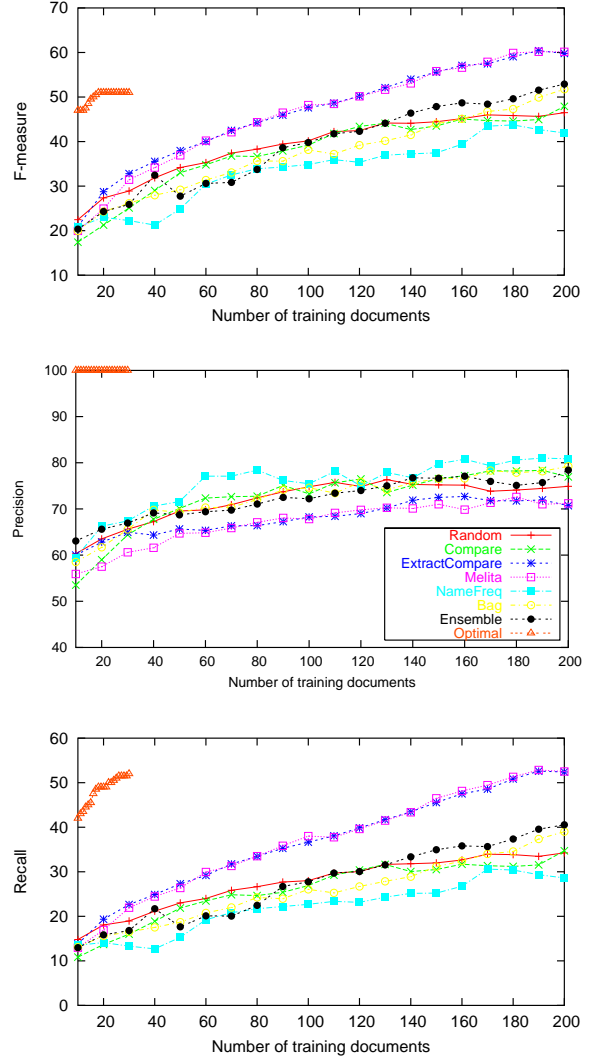


Figure 2: Learning curve for the ACQ dataset.

used to improve recall performance, while others can be used to improve precision performance. Other strategies that perform closer to random don't offer significant improvements in either precision or recall.

4.2 Reuters acquisitions articles

The ACQ dataset consists of 300 Reuters articles describing corporate acquisitions [8]. The task is to extract the name of the purchasing and acquired companies.

Fig. 2 shows the learning curves for the various se-

lection strategies on this dataset. In this case, the results are somewhat more clear cut. When looking at F1, MELITA and EXTRACTCOMPARE are significantly better than the other strategies. NAMEFREQ is again the worst. On this task, the difference in recall performance is large enough to be reflected as a large difference in the F1 performance. The boost in recall using these strategies is greater than the resulting drop in precision. As on the SA dataset, when precision is considered, NAMEFREQ performs best, with MELITA and EXTRACTCOMPARE performing worst. The relative performance of the selection strategies is reversed


```

-----
KEY CENTURION [KEYC] COMPLETES ACQUISITIONS
CHARLESTON, W.Va., April 2 - Key Centurion Bancshares Inc said it has
completed the previously-announced acquisitions of Union Bancorp of West
Virginia Inc and Wayne Bancorp Inc.
Reuter
-----
HCI & G SEMINAR
Wednesday, January 13, 1993
3:30 - 5:00pm
Wean Hall 5409

Aiding Performance in Complex Dynamic Worlds:
Some HCI Challenges

Emilie Roth
Information Technology Dept.
Westinghouse Science and Technology Center

We have been studying crew problem-solving and decision-making in
simulated power plant emergencies with the objective of developing the
next generation computerized control room. Power plant control rooms
offer some unique challenges to HCI. Because they are complex ...
-----

```

Figure 3: The most-informative ACQ (top) and SA (bottom) documents.

when we consider precision instead of recall. The two strategies that perform best when recall is considered are those that perform worst when precision is considered.

Again this indicates that the various strategies are suited to optimizing either precision or recall. Given this trend, we investigate whether selecting documents according to both kinds of strategy will improve both precision and recall. The ensemble selection strategy selects documents according to both MELITA (improves recall) and NAMEFREQ (improves precision). This approach performs slightly better than random for both precision and recall, but not as well as NAMEFREQ for precision or MELITA for recall.

4.3 Discussion

For each task, we have shown the first few points of the optimal learning curve. On each task, the optimal curve is several times better than the best selection strategy in the early stages of learning. This indicates that there is plenty of scope for improved selection strategies. Indeed the optimal curve shows that the choice of initial training documents can lead to very good performance. For example, on the SA dataset there is a single document (see Fig. 3) that when the learner is trained on, it performs with F1 of 24.25% on the rest of the training corpus. On the ACQ dataset, there is a single document that gives an F-score of 21.5%. On the SA dataset, best performing strategy (MELITA) requires 130 documents to achieve the same performance as the optimal after 20 documents. On the ACQ dataset, MELITA requires 130 documents to achieve the same F1 performance as the

optimal strategy after 30 documents. For recall, it requires 190 documents to achieve the same performance as the optimal recall strategy. Even after 200 documents it does not reach the level of performance of the optimal precision curve. This indicates that there are a small number of highly informative examples in the dataset, while all the other documents contribute only very small incremental increases in performance.

There is clear trade-off between optimizing precision, recall or F1. Fig. 4 shows the learning curves when optimizing for F1, precision and recall respectively for the ACQ dataset. The optimal precision curve results in low recall, and vice-versa. This trend is to be expected, but Fig. 4 shows that the trade-off is not complete. While we can maximize precision at 100% if we are prepared to accept very low recall, the optimal recall curve is much lower. We cannot achieve very high recall, even if we are prepared to accept very low precision. We conjecture that this is because, as a covering algorithm, LP^2 is inherently biased to favor precision over recall.

The choice of strategy depends on whether we wish to optimize for precision or recall. We have shown that some strategies perform better than random at improving precision, while others perform better at improving recall.

Given that MELITA improves recall and NAMEFREQ improves precision, we attempted to improve both by combining both approaches. However this ENSEMBLE approach does not perform as well as either approach.

5 Predicting performance

The previous experiments concerned the relative performance of the selection strategies. From a practical perspective, it is important to be able to predict which strategy will perform best, without having to actually try the strategies and measure the results. We now turn to some preliminary results that address this issue.

In order to predict the relative performance of the different selection strategies, we need to find some informative property of the strategies that can be measured without knowing the labels of the unlabeled data. We have used the correlation between the documents selected by each strategy. Our hypothesis is that if two strategies tend to select the same documents, then they will have similar performance, while if two strategies select very different documents, then there will be a large performance gap between the two. Our ultimate

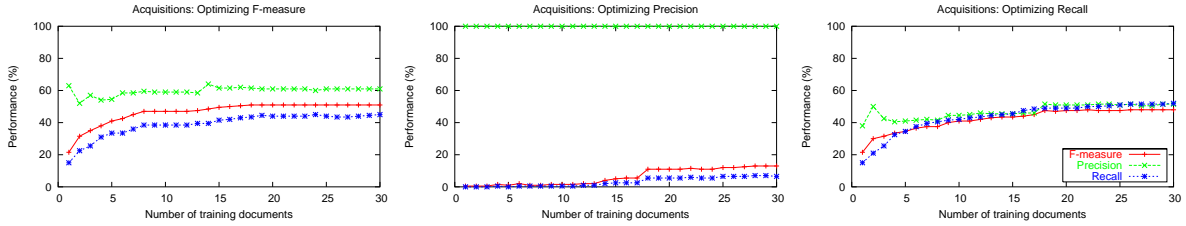


Figure 4: Optimal learning curves for F1, precision and recall on the ACQ dataset.

goal is to derive such a relationship analytically. We now consider empirical evidence that partially supports this hypothesis.

To measure the degree of agreement between two strategies, we first randomly select 50 documents. Then, in batches of 10, we selected the remaining documents using each selection strategy. This was repeated 10 times and the average Spearman rank correlation coefficient calculated for each pair of strategies. Strategies that select documents in the same order have a correlation of $+1$, while strategies that select documents in the opposite order have a correlation of -1 .

On both tasks, there is a strong positive correlation between EXTRACTCOMPARE and MELITA, indicating that they both tend to pick the same documents. There is also a positive correlation between ENSEMBLE and MELITA and NAMEFREQ. This is expected as ENSEMBLE combines these two strategies.

On the SA task, there is quite a strong negative correlation between NAMEFREQ and MELITA. There is a slight negative correlation between these strategies on the ACQ dataset. This indicates that these strategies tend to select different documents.

To determine whether selection agreement is useful for predicting relative performance, we then measured the performance gap between the strategies. We define $\text{gap}(x, y)$ as the normalized performance difference, averaged over all points on the learning curve from 50 to 200 documents.

Fig. 5 shows the selection agreement between various selection strategy pairs plotted against the gap in performance between the strategies. We display SA and ACQ in different plots, and we measure the gap in precision, recall and F1. Anecdotally, it is apparent that our ability to predict the performance gap is quite good for strategies that are highly correlated (either positively or negatively), but rather poor when the strategies are weakly correlated.

More precisely, our hypothesis that selection agree-

ment can be used to predict performance gap is validated to the extent that these data have a correlation of -1 . Fig. 6 shows the six correlations. As anticipated, all of the correlations are negative, though weakly so. Our approach is slightly better at predicting the performance gap for SA compared to ACQ, and for predicting the recall gap compared to precision and F1.

6 Conclusion

We have investigated several Active Learning selection strategies that can be applied to Information Extraction. Of these, several performed significantly better than a random selection strategy. MELITA and EXTRACTCOMPARE offer improved recall over random selection with a resulting drop in precision. NAMEFREQ offers improved precision at the expense of recall. Some strategies offer improvements in recall while others improve precision, but it is difficult to get significant improvement in both recall and precision. Most importantly, there is still however a significant difference in performance between the optimal curve and the various selection strategies. Existing selection strategies still have significant scope for improvement.

Our immediate future work involves identifying strategies that bridge the wide gap between the optimal strategy and the strategies we have investigated so far. For example, we are exploring a committee-based strategy called DUAL that has two committee members for each field: one that extracts the field itself, and one that extracts all document fragments except the particular field. We are also conducting a detailed analysis of the optimal documents to determine strategies that can bridge the gap.

A second goal is to improve our ability to predict the performance gap between two strategies. Ultimately, we seek a theoretically-grounded model of active learning that will enable us to derive upper or lower bounds

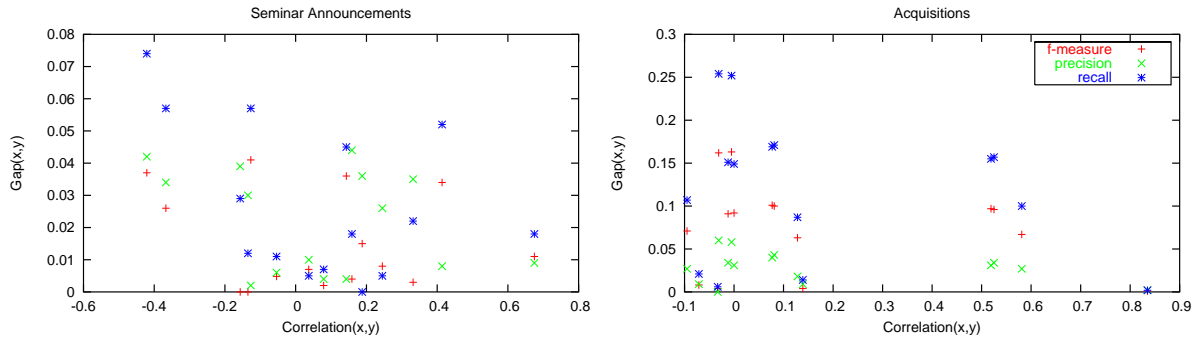


Figure 5: Performance gap vs. selection correlation.

	F1	P	R	mean
SA	-0.27	-0.32	-0.43	-0.34
ACQ	-0.22	-0.25	-0.23	-0.23
mean	-0.25	-0.29	-0.33	

Figure 6: The correlation between two strategies' performance gap and the degree to which they select the same documents.

on the performance of a given strategy.

Acknowledgements

This research was supported by grants SFI/01/F.1/C015 from Science Foundation Ireland, and N00014-03-1-0274 from the US Office of Naval Research. We thank Fabio Ciravegna for access to LP².

References

- [1] M. Califf and R. Mooney. Relational learning of pattern-match rules for information extraction. In *Proc. 16th Nat. Conf. Artificial Intelligence*, 1999.
- [2] F. Ciravegna. Adaptive information extraction from text by rule induction and generalisation. In *Proc. 17th Int. Joint Conf. Artificial Intelligence*, 2001.
- [3] F. Ciravegna, A. Dingli, D. Petrelli, and Y. Wilks. Timely and non-intrusive active document annotation via adaptive information extraction. In *ECAI Workshop Semantic Authoring Annotation and Knowledge Management*, 2002.
- [4] F. Ciravegna, A. Dingli, D. Petrelli, and Y. Wilks. User-system cooperation in document annotation based on information extraction. In *13th International Conference on Knowledge Engineering and Knowledge Management*, 2002.
- [5] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2), 1994.
- [6] I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *International Conference on Machine Learning*, 1995.
- [7] D. Freitag. *Machine Learning for Information Extraction in Informal Domains*. PhD thesis, Carnegie Mellon University, 1998.
- [8] D. Freitag. Toward general-purpose learning for information extraction. In *35th Annual Meeting of the Association for Computational Linguistics*, 1998.
- [9] D. Freitag and N. Kushmerick. Boosted wrapper induction. In *Proc. 17th Nat. Conf. Artificial Intelligence*, 2000.
- [10] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *11th International Conference on Machine Learning*, 1994.
- [11] I. Muslea, S. Minton, and C. Knoblock. Selective sampling with redundant views. In *Proc. 17th Nat. Conf. Artificial Intelligence*, 2000.
- [12] T. Scheffer and S. Wrobel. Active learning of partially hidden Markov models. *Active Learning, Database Sampling, Experimental Design: Views on Instance Selection*, 2001.
- [13] C. Thompson, M. Califf, and R. Mooney. Active learning for natural language processing and information extraction. In *Proc. 16th Int. Conf. Machine Learning*, 1999.

Active Learning for Information Extraction with Multiple View Feature Sets

Rosie Jones

ROSIE.JONES@CS.CMU.EDU

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Rayid Ghani

RAYID.GHANI@ACCENTURE.COM

Accenture Technology Labs, 161 N. Clark St, Chicago, IL 60601 USA

Tom Mitchell

TOM.MITCHELL@CS.CMU.EDU

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Ellen Riloff

RILOFF@CS.UTAH.EDU

School of Computing, University of Utah, Salt Lake City, UT 84112 USA

Abstract

A major problem with machine learning approaches to information extraction is the high cost of collecting labeled examples. Active learning seeks to make efficient use of a labeler's time by asking for labels based on the anticipated value of that label to the learner. We consider active learning approaches for information extraction problems where each example is described by two distinct sets of features, either of which is sufficient to approximate the function; that is, they fit the cotraining problem setting. We discuss a range of active learning algorithms and show that using feature set disagreement to select examples for active learning leads to improvements in extraction performance regardless of the choice of initially labeled examples. The result is an active learning approach to multiple view feature sets in general, and noun phrase extraction in particular, that significantly reduces training effort and compensates for errors in initially labeled data.

1. Introduction

One difficulty with machine learning techniques for information extraction is the high cost of collecting labeled examples. We can make more efficient use of the trainer's time by asking them to label only instances that are most useful for the learner. Research in active

learning has shown that using a pool of unlabeled examples and prompting the user to only label examples that have high anticipated value reduces the number of examples required for tasks such as text classification, parsing, and information extraction (Thompson et al., 1999; Soderland, 1999). Bootstrapping algorithms have been proposed for similar learning problems (Blum & Mitchell, 1998; Nigam & Ghani, 2000; Collins & Singer, 1999; Muslea et al., 2000) that fall into the cotraining setting *ie.* they have the property that each example can be described by multiple feature sets, any of which are sufficient to approximate the function.

We present an active learning framework for problems that fall into the cotraining setting. Our approach differs from work by Muslea on co-testing (Muslea et al., 2000) in that semi-supervised learning, using both labeled and unlabeled data, is interleaved in the active learning framework. Instead of learning with only labeled data, we use a bootstrapping algorithm to learn from both labeled and unlabeled data and then select examples to be labeled by the user at every iteration. We do this by adapting co-EM to information extraction tasks and develop active learning techniques that make use of multiple feature sets.

We focus on extracting noun phrases that belong to a predefined set of semantic classes. Using the words within the noun phrase and the words surrounding it as two distinct feature sets, we describe active learning algorithms that make effective use of this division. Instead of relying only on a fixed, prelabeled set of ex-

amples, the active learning system keeps the user in the loop and presents them with examples to label at each iteration. We find that by utilizing the redundancy inherent in the data because of multiple feature sets, active learning approaches can significantly reduce the effort required to train information extraction systems. We also show that active learning can compensate for a bad choice of initial labeled examples and that the labeling effort is better spent *during* the active learning process rather than at the beginning, as done in standard supervised and semi-supervised learning.

2. Task and Data Set Representation

We focus on extracting noun phrases that correspond to organizations, people, and locations from a set of web pages. These semantic classes are often identified using *named entity recognizers* (e.g. (Collins & Singer, 1999)), but those tasks are usually limited to proper names, such as “John Smith” or “California”. Our task is to identify all relevant noun phrases, such as “a telecommunications company” or “software engineer”. As our data set, we used 4392 web pages from corporate websites collected for the WebKB project (Craven et al., 1998). 4160 were used for training and 232 were set aside as a test set. We preprocessed the web pages by removing HTML tags, and adding periods to the end of sentences when necessary. To label our data set, we extract all noun phrases (NPs) and manually label them with one or more of the semantic classes of interest (organizations, people, or locations). If a noun-phrase does not belong to any of these classes, it is assigned *none*.

Our goal is to recognize the semantic class of a word or phrase *in context*. Many words can belong to different semantic classes when they appear in different contexts. For example, the word “leader” can refer to a person, as in “a number of world leaders and other experienced figures” but can also occur in phrases which do not represent people, such as in “the company is a world leader”. We have identified three common situations where the semantic class of a word can vary depending on the local context:

- **General Polysemy:** many words have multiple meanings. For example, “company” can refer to a commercial entity or to companionship.
- **General Terms:** many words have a broad meaning that can refer to entities of various types. For example, “customer” can refer to a person or a company.
- **Proper Name Ambiguity:** proper names can be associated with entities of different types. For exam-

ple, “John Hancock” can refer to a person or a company, which reflects the common practice of naming companies after people.

Although the semantic category of a noun phrase may be ambiguous, the context in which the noun phrase occurs is often sufficient to resolve its category. Therefore, we cast our problem as one of classifying each instance of a noun phrase that appears within a document, based on both the noun phrase and its surrounding context. Each *noun phrase instance* (or example) consists of two items: (1) the noun phrase itself, and (2) a lexico-syntactic context surrounding the noun phrase. We used the AutoSlog (Riloff, 1996) system to generate patterns representing the lexico-syntactic contexts. For the remainder of the paper we will refer to these lexico-syntactic contexts simply as *contexts*.

By using both the noun phrases and the contexts surrounding them, we provide two different types of features to our classifier. In many cases, the noun phrase itself will be unambiguous and clearly associated with a semantic category (e.g., “the corporation” will nearly always be an organization). In these cases, the noun phrase alone would be sufficient for correct classification. In other cases, the context itself is highly predictive. For example, the context “subsidiary of <>” nearly always refers to an organization. In those cases, the context alone is sufficient. There will also be cases where either of these features by itself will be ambiguous with respect to the semantic class. We discuss these ambiguities in the next section and measure the extent to which these are present in our data set.

2.1. Ambiguity of Classes

Since our training corpus is unlabeled, we cannot measure the ambiguity directly. However, the examples in our test set were randomly drawn from the same distribution, and manually labeled, so ambiguity of noun-phrases and contexts in the test set is indicative of their ambiguity in the training set as well. During the labeling process, when an example was judged as belonging to multiple classes, multiple labels were assigned. An example is the sentence “We welcome feedback”, where the “We” could refer to an organization, or the people of the organization. This kind of ambiguity also occurs when countries (locations) act as agents (organizations). Tables 1 and 2 summarize the ambiguity in noun-phrases and contexts in our test set, by showing how many noun-phrases were ambiguous with respect to which classes. Noun-phrases that did not fall into any of the categories *location*, *organization* or *person* were labeled as *none*.

Noun phrases are mostly unambiguous (only 2% of the

Ambiguity	Class(es)	Number of NPs
No Ambiguity	none	3574
	loc	114
	org	451
	person	189
Belonging to TWO classes	loc, none	6
	org, none	31
	person, none	25
	loc, org	6
	org, person	13
Belonging to THREE classes	loc, org, none	1
	org, person, none	3

Table 1. Noun-Phrase Ambiguity: The number of NPs that belong to each combination of classes. NPs are relatively unambiguous (4328 out of 4413 only belong to a single class).

Ambiguity	Class(es)	Number of Contexts
No Ambiguity	none	1068
	loc	25
	org	98
	person	59
Belonging to TWO classes	loc, none	51
	org, none	271
	person, none	206
	loc, org	5
	org, person	50
Belonging to THREE classes	loc, org, none	18
	org, person, none	83
Belonging to all FOUR classes	loc, org, person, none	6

Table 2. Context Ambiguity: The number of Contexts that belong to each combination of classes. Contexts are relatively ambiguous - 6 of the contexts were labeled as belonging to all 4 classes

4413 unique noun-phrases belong to 2 or more classes) but are relatively sparse in the training set; only 1887 of these noun-phrases had been seen in the training set. Thus for 57% of the noun-phrases in the test set, we have no training information at all. In contrast, 37% of the contexts are ambiguous but each of them occurs more often and can be modeled better. 91% of these 1940 contexts from the test set also appear in the training set. These measurements reinforce our previous assumption both the noun phrase and the context will play a role in determining the correct classification for each example.

3. Active Learning Problem Setting

Active learning is the problem of determining which unlabeled instances to label next as learning proceeds, in order to learn most accurately from the least labeling effort. The detailed problem setting varies with

the form of the target function, the pool of unlabeled instances available, and the type of training information sought from the trainer. This section defines the active learning problem setting we consider, by placing it along each of these dimensions.

Form of target function to be learned. We consider active learning of a target function $f : X \rightarrow Y$ that maps a set X of instances to a set Y of possible values. We only consider target functions where instances are described by two distinct sets of features X_1 and X_2 (i.e., $X = X_1 \times X_2$), such that the target function can be approximated either in terms of X_1 or in terms of X_2 . In our information extraction task, X_1 describes the noun phrase itself, and X_2 describes the context in which it appears.

For example, consider a problem where each instance $x \in X$ is a noun phrase along with its surrounding linguistic context (e.g., “drove to (New York)”), where the target function f specifies whether or not the noun phrase refers to a location, where the first set of features X_1 consists of the noun phrase itself (e.g., (New York)), and where the second set of features X_2 consists of the linguistic context (e.g., “drove to ()”). Ideally, we assume that the target function f can be expressed in terms of X_1 alone, and also in terms of X_2 alone (e.g., that it is possible to determine whether the instance refers to a location, based solely on the context “I drove to ()”, and also based solely on the noun phrase “(New York).” Put more formally, we assume $X = X_1 \times X_2$, where there exist functions $g_1 : X_1 \rightarrow Y$ and $g_2 : X_2 \rightarrow Y$ such that $f(x) = g_1(x_1) = g_2(x_2)$ for all $x = x_1|x_2$. In the real-world domains considered here, this ideal assumption is not fully satisfied, as described in Section 2 and Tables 1 and 2.

Pool of unlabeled data available. A second dimension for defining the active learning problem involves assumptions about how and when unlabeled instances are made available to the active learner. We begin with the usual PAC-learning assumption (Ehrenfeucht et al., 1989), that the instances X are generated according to some fixed but unknown probability distribution $P(X)$, and that the goal of the learner is to minimize the probability that it will misclassify future instances drawn randomly according to this same distribution. We can make several assumptions about how unlabeled instances are obtained by the active learner. We could assume that a fixed pool containing n instances is collected in advance according to the distribution $P(X)$, and that this fixed pool is all that is available to the active learner. This setting is considered in (McCallum & Nigam, 1998). We call this the *fixed random pool* setting. An alternative is to as-

sume that the active learner can draw new instances at random from $P(X)$ during learning, so that it is not limited to the fixed pool. This setting is considered in (Cohn et al., 1994), and we will refer to this as the *on-going random sampling* setting. A further alternative is to assume the learner can synthesize any syntactically legal instance in X , regardless of $P(X)$, and ask the trainer for information about this instance. While this setting is interesting, it can lead to synthetic examples that are not intelligible to the trainer (Baum & Lang, 1992). This setting is considered in (Angluin, 1988). We will call this the *synthesized instances* setting, though it has sometimes been referred to as “membership querying.” In this paper, we consider only the fixed random pool setting for active learning.

Information provided by the trainer. A third dimension concerns what information is to be provided to and by the trainer. In the *standard labeling* setting the trainer is provided an unlabeled instance, and in return provides the label. A different possibility is that the trainer is provided part of the description (e.g., provided only “drove to $\langle \rangle$ ”), and required to label it. We will call this the *single feature set labeling* setting. Another possibility is that the trainer is allowed to demur in some cases, providing a label only when certain (e.g., the trainer may decline to label “occurred in $\langle \rangle$ ” because of its ambiguity, but agree to label “drove to $\langle \rangle$ ” as a reliable location context).

To summarize, we consider an active learning problem in which the target function follows the *cotraining assumption*, the data available to the active learner is a *fixed random pool*, and we compare *standard labeling* to *single feature set labeling*.

4. Algorithmic Overview

Our approach consists of the following steps: a small set of words (*seedwords* or *seeds*) and a set of documents are provided. Instances in the document collection are initially labeled using the seeds (*initial examples*) and the annotated documents (including the unlabeled instances) are given to the bootstrapping algorithm. After every iteration of the bootstrapping algorithm, a human labeler is asked to label a set of examples selected by the active learning method. The design of our information extraction system requires answering the following questions:

1. How to label the initial examples for the bootstrapping algorithm?
2. What bootstrapping method will be used to learn from a combination of labeled and unlabeled data?

Class	SeedWords
locations	australia, canada, china, england, france, germany, japan, mexico, switzerland, united states
organizations	inc., praxair, company, companies, arco dataram, halter marine group, xerox, rayonier timberlands, puretec
people	customers, subscriber, people, users, shareholders, individuals, clients, leader, director, customer

Table 3. Seedwords used for initialization of bootstrapping.

3. What is the best active learning algorithm for requesting additional labels from the trainer?
4. What is the best method to assign labels to *test* instances?

We discuss and answer these questions below.

4.1. Method for Initial Labeling

The set of seedwords we use to generate initially labeled examples for the three information extraction tasks are shown in Table 3. The *locations* seedwords are the same as those used in (Riloff & Jones, 1999). The seeds for *organizations* and *people* were chosen by sorting noun-phrases in the training set by frequency, and selecting the first ten matching the target class. Note that this method does not necessarily lead to the best choice of seedwords, but is a simple method not requiring skill and experience. A domain expert might be able to pick better seedwords but we wanted to experiment with words that a non-expert could easily generate. Seedwords may be of poor quality if they are either (1) infrequent in the documents, or (2) ambiguous. We run experiments to examine whether active learning can compensate for “poor” seedwords (in terms of both 1 and 2) and report results in Section 5.

There was ambiguity across all sets of seedwords. In particular, “leader” refers to an organization more often than to a person in our data set, but it was used as a person keyword during Fixed Initialization. We will show that our algorithms are robust enough to recover from this kind of ambiguity. We examine two methods for creating initial labeled examples to jumpstart the bootstrapping process, one of which allows us to correct these ambiguities at the beginning:

Fixed Initialization: All noun phrases whose head noun (right-most word) matches a seed word are considered to be positive training instances, regardless of the context in which they appeared. This approach was also used in (Riloff & Jones, 1999). This is frequently correct, for example labeling the city “Columbia” as a location in the example “... head-

quartered in Columbia”. However, in the example “Columbia published ...” it refers to a publishing company, not a location.

Active Initialization: To address errors introduced by ambiguity in the automatic labeling phase, we tried a novel method that incorporates active learning. In *active initialization*, examples matching the seed words are interactively labeled by the trainer before beginning the bootstrapping process. We hypothesize that by actively labeling examples at the outset and correcting the errors introduced by ambiguous seedwords, we can provide the bootstrapping algorithms with better initial examples and thus improve extraction performance. For reasonably frequent seed words, this requires significant numbers of examples to be labeled at the outset; 669 examples for *locations*, 3406 for *organizations*, and 2521 for *people*, for the seed words in Table 3 and our training collection.

4.2. Bootstrapping Method: coEM

Unlike previous work in active learning where the classifiers are only learned on labeled data, we use a bootstrapping method to learn from both labeled and unlabeled data. The bootstrapping algorithm we use for the information extraction task is coEM. *coEM* is a hybrid algorithm, proposed by (Nigam & Ghani, 2000), combining features from both co-training and Expectation-Maximization (EM). coEM is iterative, like EM, but uses the feature split present in the data, like co-training. The separation into feature sets we used is that of noun-phrases and contexts. The algorithm proceeds by initializing the noun-phrase classifier $\hat{g}_1(x_1)$ using the labeled data only. Then $\hat{g}_1(x_1)$ is used to probabilistically label all the unlabeled data. The context classifier $\hat{g}_2(x_2)$ is then trained using the original labeled data plus the unlabeled data with the labels provided by \hat{g}_1 . Similarly, \hat{g}_2 then relabels the data for use by \hat{g}_1 , and this process iterates until the classifiers converge. For final predictions over the test set, \hat{g}_1 and \hat{g}_2 predictions are combined by assuming independence, and assigning the test example probability proportional to $\hat{g}_1(x_1)\hat{g}_2(x_2)$.

In earlier work (Ghani & Jones, 2002), we compared coEM with metabootstrapping (Riloff & Jones, 1999) and found coEM to be better.

4.3. Active Learning Methods in the Cotraining Setting

The cotraining problem structure lends itself to a variety of active learning algorithms. In *co-testing* (Muslea et al., 2000) the two classifiers are trained only on available labeled data, then run over the unlabeled

data. A *contention set* of examples is then created, consisting of all unlabeled examples on which the classifiers disagree. Examples from this contention set are selected at random, a label is requested from the trainer, both classifiers are retrained, and the process repeats.

While this naïve co-testing algorithm was shown to be quite effective, it represents just one possible approach to active learning in the co-training setting. It is based on training the two classifiers \hat{g}_1 and \hat{g}_2 using labeled examples only, whereas work by (Collins & Singer, 1999; Blum & Mitchell, 1998; Riloff & Jones, 1999) has shown that unlabeled data can bootstrap much more accurate classifiers. In this paper, we use both labeled and unlabeled data to create our classifiers. Also, instead of selecting new examples uniformly at random from the contention set, one might rank the examples in the contention set according to some criterion reflecting the value of obtaining their label. In this paper, we propose and experiment with active learning algorithms that use unlabeled data for training \hat{g}_1 and \hat{g}_2 , in addition to using \hat{g}_1 and \hat{g}_2 to determine which unlabeled example to present to the trainer. We also consider a variety of strategies for selecting the best example from the contention set:

Uniform Random Selection: This baseline method selects examples according to a uniform distribution. Each noun-phrase, context pair that occurs at least once in the training set is selected with equal probability. Example frequency is ignored.

Density Selection: The most frequent unlabeled noun-phrase context pair is selected for labeling at each step. This method is based on the assumption that labeling frequently occurring examples would be beneficial for the learner.

Feature Set Disagreement: Since we learn two distinct classifiers that apply to the same instance, one way to select instances where a human trainer can provide useful information is to identify instances where these two classifiers disagree. This approach can be viewed as a form of *query-by-committee* (QBC), (Freund et al., 1997; Liere & Tadepalli, 1997) or *uncertainty sampling* (Thompson et al., 1999), where the committee consists of models that use different feature sets and is similar to that used by (McCallum & Nigam, 1998). Our selection criterion is based on Kullback-Leibler (KL) divergence. Our final ranking gave each example a density-weighted KL score, by multiplying $KL(\hat{P}_{g_1}(+|x), \hat{P}_{g_2}(+|x))$ by the frequency of the example. Examples were selected deterministically, with the next unlabeled example taken each time. For these experiments we used only a single

committee member per feature set, though an obvious extension is to have multiple committee members per feature set.

Context Disagreement: All *active* selection algorithms described so far use the *standard labeling* paradigm, with the user labeling a pair consisting of a noun-phrase and its context. However, we can also label noun phrases independent of context, and since each noun phrase may occur in many contexts, this may lead to greater value in labeling. For example, “Italy” occurs with “centers in ⟨⟩”, “operations in ⟨⟩”, “introduced in ⟨⟩”, “partners in ⟨⟩”, and “offices in ⟨⟩”, so labeling “Italy” provides us with information about all of these contexts. In addition, we can use the different contexts as votes by committee members about the label for the noun-phrase. Selecting the noun-phrase with the most *context disagreement* may provide the bootstrapping algorithm with the most informative labeling. This can be thought of as query-by-committee (QBC) with the committee consisting of different cooccurrences of elements of one feature set with elements of the other feature set. We quantified context disagreement using density weighted KL divergence to the mean, as in feature set disagreement, and all the contexts of the noun-phrase were used as input to the KL divergence measure. We used the frequency of the noun-phrase to density-weight the KL divergence. The user then only labeled noun-phrases, in *single-feature set labeling*.

4.4. Extraction Method

The combination of bootstrapping and active learning results in a learned model consisting of noun-phrases and contexts, with corresponding learned probabilities for each. We use this model to assign scores to the unseen test instances by taking the product of the scores of the noun-phrase and context (both from the training set). Nouns and contexts that occur in the test set but have not been seen in the training set are assigned a prior to reflect the frequency of the classes in our dataset (0.027 for *locations*, 0.11 for *people* and 0.20 for *organizations*). Note that our examples include pronouns and other anaphoric references.

5. Results

We use coEM to label five examples per iteration, until 500 examples have been labeled. Then, we continue running coEM till convergence (usually around 400 iterations total) and use the learned model to score the test instances. We sort the test instances according to the score assigned by the extraction method and calculate precision-recall values.

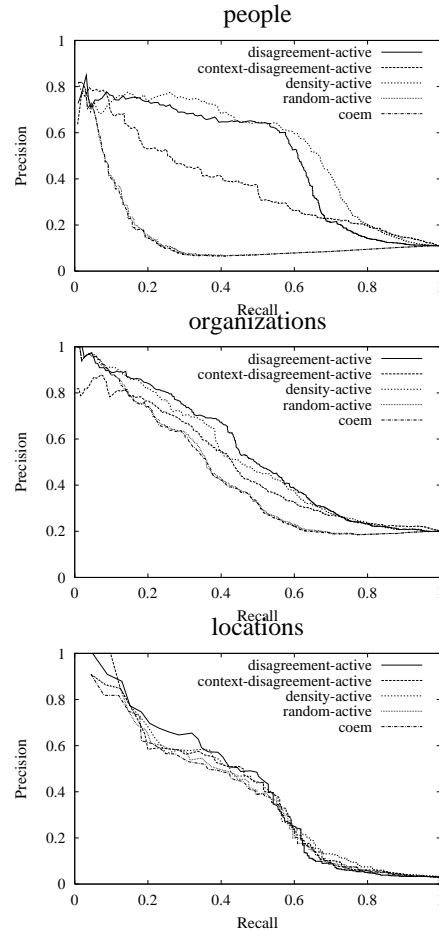


Figure 1. Comparison of active learning methods, for *locations*, *people* and *organizations*. for *people*, the sparsest class. Choosing examples to label based on disagreement between the two feature sets was most effective.

Disagreement and Density Active Learning Most Effective:

Figure 1 shows how the different active learning methods perform after the user has labeled 500 examples. Feature set disagreement (“Disagreement”) outperforms *all* other methods, except in the “people class” where density based active learning performs best. The people class contains many pronouns, which are frequently selected by density-based selection. Uniform random labeling of 500 examples does not improve over the baseline coEM using only the initial seeds. We believe this result to be significant as it shows that *randomly* selecting examples to label is no better than not labeling at all and letting coEM learn from the more promising initial labeled and unlabeled examples. This makes sense in our setting, since our positive classes are sparse, and random labeling does not provide much information about the positive class, compared with the dense information provided with the seeds. However, if we started with no seeds at all, some information would be gained by random labeling.

Although our active learning algorithms improve over the baseline in all cases, the improvement is most marked for the *people* class – this was the class with

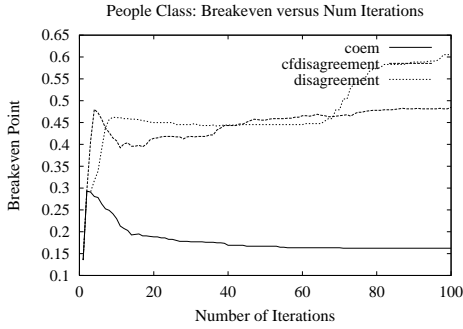


Figure 2. Breakeven point between precision and recall shown for each iteration for learning the people class. 5 examples are labeled per iteration for active learning, up to a total of 500 examples labeled. coEM improves slightly with the first few iterations, then degrades. With active learning, the most substantial improvements are made in the first few iterations, but labeling more data continues to improve results.

the most ambiguous initial seeds. This provides evidence that active learning can compensate for poor seed choices.

Labeling instances based on their frequency (“Density”) was also very effective for **people** and **organizations**, which were frequent in our dataset. Context disagreement, which uses the single feature set labeling setting, did not perform as well as Disagreement or Density labeling, which use standard labeling. Single feature set labeling is useful (better than no active learning), but using both feature sets to select instances is a more effective technique.

Substantial Improvements with Small Amounts of Labeling: As can be seen in Figure 2, the most substantial improvements with active learning are made in the first few iterations, but labeling more data continues to improve results. This suggests that we can perform favorable trade-offs between labeling time and desired levels of accuracy.

Active Learning More Useful than Active Initialization: We found that active initialization (manually labeling and correcting errors in the initial labeled examples due to ambiguous seedwords) did not perform significantly better than fixed initialization. When our active learning method is provided a set of initial instances that are “clean” and unambiguous, the extraction performance does not improve. This suggests that the active learning methods are robust to ambiguous/noisy training data and can recover from poor initial seeds. This is shown in Figure 3. We also find that the active learning method (with 500 examples labeled for **locations**) performed better than using bootstrapping with coEM with active initialization

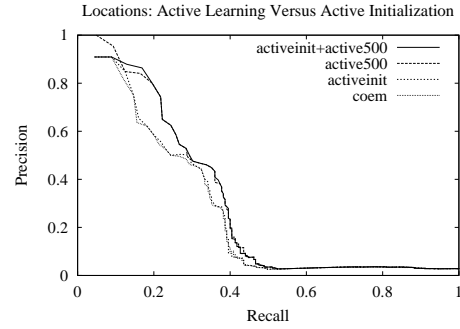


Figure 3. Labeling 500 noun-phrase-context pairs with active learning (and no initial labeling) does better than using active initialization, which requires labeling 693 examples at the outset. Labeling the initial examples and then labeling another 500 with active learning performs best.

(with 693 examples labeled). This is an important result since if we have a fixed amount of time to label instances, active learning can be a more effective use of this time than labeling the instances at the outset.

Active Learning Compensates for Infrequent Seeds: Figure 4 shows that selecting 20 country names uniformly from a set of 253 leads to variable results, with effectiveness related to frequency of initial seeds. The graph on the left contains three sets of 20 seeds each, matching 133, 129 and 34 examples in our training set, respectively. In the left-hand graph, the seed set matching only 34 examples performs poorly with the bootstrapping algorithm, but when combined with active learning, is able to produce results comparable with the other seed sets. The graph on the right shows the results for a frequent seed set (occurring 673 times in the training set). It is interesting that active learning improves results in all cases and compensates for seeds that are infrequent in the document set. Thus with active learning we can obtain results superior to bootstrapping on the best seed set, regardless of the seed set we choose.

Summary of Results We compared different metrics for selecting examples to label and found that using the disagreement between classifiers built with the two feature sets worked well. Manually correcting initial examples that were mislabeled due to ambiguous seeds is not as effective as providing the active learning algorithm with an arbitrary set of seeds and labeling examples during the learning process. Context-disagreement used the single feature set labeling setting, and did not perform as well as methods using standard labeling. Using only a single feature set for labeling may allow inaccuracies to creep into the labeled set, if any of the examples are ambiguous with respect to that feature set. In addition, disagreement

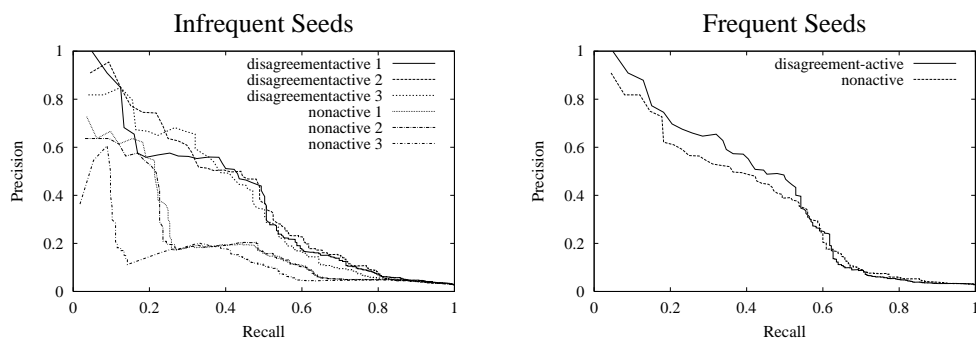


Figure 4. Active learning provides gains even with a good choice of frequent seeds (right hand graph). With a very poor set of initial seeds, active learning permits comparable results (left hand graph).

between members of a single feature set may reflect inherent ambiguity in the example, and not uncertainty in the learner.

6. Conclusions

We presented a framework for incorporating active learning in the semi-supervised learning paradigm by interleaving a bootstrapping algorithm that learns from both labeled and unlabeled data with a variety of sample selection techniques that present the user with examples to label at each iteration. We show that employing the redundancy in feature sets and designing algorithms that exploit this redundancy enables both bootstrapping and active learning to be effective for training information extractors. The techniques presented in this paper are shown to be robust and are able to compensate for bad choice of initial seedwords. Although the results shown here are specific to the information extraction setting, our approach and framework are likely to be useful in designing active learning algorithms for settings where a natural, redundant division of features exists.

References

- Angluin, D. (1988). Queries and concept learning. *Machine Learning*, 2, 319–342.
- Baum, E., & Lang, K. (1992). Query learning can work poorly when a human oracle is used. *International Joint Conference on Neural Networks*.
- Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *COLT-98*.
- Cohn, D. A., Atlas, L., & Ladner, R. E. (1994). Improving generalization with active learning. *Machine Learning*, 15, 201–221.
- Collins, M., & Singer, Y. (1999). Unsupervised Models for Named Entity Classification. *EMNLP/VLC*.
- Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., & Slattery, S. (1998). Learning to Extract Symbolic Knowledge from the World Wide Web. *AAAI-98*.
- Ehrenfeucht, A., Haussler, D., Kearns, M., & Valiant, L. (1989). A general lower bound on the number of examples needed for learning. *Information and Computation*, 82, 247–261.
- Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, 28, 133–168.
- Ghani, R., & Jones, R. (2002). A comparison of efficacy of bootstrapping algorithms for information extraction. *LREC 2002 Workshop on Linguistic Knowledge Acquisition*.
- Liere, R., & Tadepalli, P. (1997). Active learning with committees for text categorization. *AAAI-97*.
- McCallum, A. K., & Nigam, K. (1998). Employing EM and pool-based active learning for text classification. *ICML*.
- Muslea, I., Minton, S., & Knoblock, C. A. (2000). Selective sampling with redundant views. *AAAI/IAAI*.
- Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. *CIKM-2000*.
- Riloff, E. (1996). An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. 85, 101–134.
- Riloff, E., & Jones, R. (1999). Learning Dictionaries for Information Extraction by Multi-level Bootstrapping. *AAAI-99*.
- Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34, 233–272.
- Thompson, C. A., Califf, M. E., & Mooney, R. J. (1999). Active learning for natural language parsing and information extraction. *ICML*.

Information Extraction from Multi-Document Threads

David Masterson and Nicholas Kushmerick

Dept. of Computer Science, University College Dublin

{david.masterson,nick@ucd.ie}

Abstract

Information extraction (IE) is the task of extracting fragments of important information from natural language documents. Most IE research involves algorithms for learning to exploit regularities inherent in the textual information and language use, and such systems generally assume that each document can be processed in isolation. We are extending IE techniques to multi-document extraction tasks, in which the information to be extracted is distributed across several documents. For example, many kinds of work-flow transactions are realized as sequences of electronic mail messages comprising a conversation among several participants. We show that IE performance can be improved by harnessing the structural and temporal relationships between documents.

1 Introduction

Information Extraction (IE) is an important approach to automated information management. IE is the task of converting documents containing fragments of structured information embedded in other extraneous material, into a structured template or database-like representation. For example, to help a financial analyst identify trends in corporate mergers and acquisitions, an IE system could convert a stream of financial news articles into a series of templates, each of which captures the details of a single transaction as reported in one news article.

The key challenges to effective IE are the inherent complexity and ambiguity of natural language, and the need for IE systems to be rapidly ported

to new domains. For example, one would hope that an IE system tuned for the mergers and acquisitions domain could be rapidly reconfigured to extract information about (for example) hostile takeovers. Machine learning has thus emerged as a powerful approach to developing adaptive information extraction systems that rapidly scale to new domains [3, 8, 10, 11, 6, 4]. The general approach is that the IE system learns extraction rules by generalizing from a set of training documents, each manually annotated with the correct information to extract.

As depicted in Figure 1, nearly all existing adaptive IE algorithms make a powerful assumption: each extracted template can be filled by analyzing a single document in isolation. In many cases this assumption is realistic, but we are interested in multi-document extraction tasks in which it is highly unrealistic.

Multi-document extraction tasks arise naturally in many work-flow scenarios, in which some operational process or transaction is realized as a “conversation” between several participants distributed over several natural language texts. We are particularly motivated by electronic mail work-flow streams [2]. For example, in Sec. 3, we evaluate our approach on a corpus of email message threads, each of which discusses a postgraduate student application, and the task is to convert each thread into the structured representation needed to actually register the student with the university authorities.

Multi-document workflow streams are important for two reasons. From an application perspective, our techniques are an enabling technology for a variety of automated logging and monitoring tools for work-flow processes that are realized as docu-

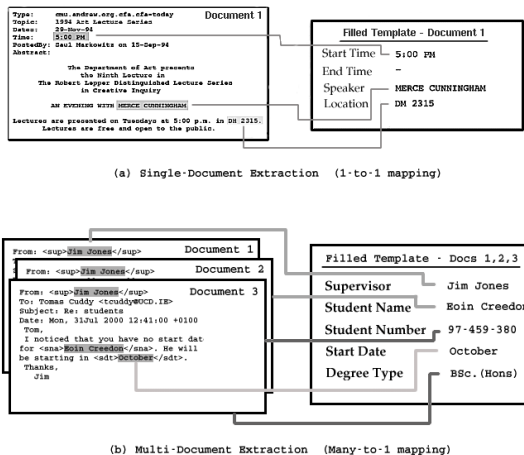


Figure 1: (a) Most IE research assumes a one-to-one correspondence between extracted templates and documents. (b) In multi-document tasks, each template draws content from multiple related documents.

ment sequences. From a research perspective, multi-document extraction is a challenging new direction for which existing techniques are inadequate.

Our approach to multi-document extraction is based on two phases. In a pre-processing phase, the training data is augmented with synthetic documents in an attempt to improve recall. We then invoke an off-the-shelf adaptive IE algorithm to learn a set of extraction rules. During extraction, these rules are invoked on a document thread, and the resulting templates are post-processed using the temporal structure of the message stream to disambiguate extraction decisions and hence improve precision. Our experiments in a challenging real-world multi-document extraction task demonstrate a 15% improvement over the core learning algorithm.

The remainder of this paper is organized as follows. Section 2 describes our approach to multi-document extraction. Section 3 demonstrates that our approach is effective in a corpus of real email. Section 4 describes related work, and Section 5 summarizes our

results and suggests future directions for our research.

2 Multi-document extraction

We begin by describing the multi-document extraction problem in more detail, and then describe our proposed approach to multi-document extraction.

2.1 Problem formulation

As in standard information extraction tasks, we assume as input a template that is to be instantiated from a set of documents. The template comprises several fields, and the goal is to extract particular fragments from the documents to assign to each field. Accuracy is measured in terms of whether the extracted values are correct, compared to a reference template provided by a human annotator.

As shown in Figure 1, traditional single-document extraction techniques assume that the template values should be drawn from a single document. In contrast, in a multi-document setting, the document collection is partitioned into sets of related documents, and a single template is constructed from each such set. In our work-flow scenario, each set corresponds to a single operational process or transaction, and the goal is to summarize the entire transaction in one template. Note that we assume that this partitioning is provided as input. We defer to future work the use of text classification and clustering algorithms to automatically partition a sequence of interleaved messages.

2.2 Approach

Our approach to multi-document extraction involves invoking an existing adaptive IE algorithm as a sub-routine. More precisely, an adaptive IE algorithm is a function `LEARN` that takes as input a set of training documents `TRAIN`, and outputs a set of extraction rules `RULES`. Informally, we describe this process using a functional notation: `LEARN(TRAIN) → RULES`. Note that the training documents `TRAIN` have been manually annotated to indicate the correct fragments.

In our experiments, we used the state-of-the-art (LP)² algorithm [4] for LEARN, but our technique is applicable to any adaptive IE algorithm. In particular, our approach does not depend on any specific rule language (indicated generically as RULES above). Indeed, our approach is suitable for IE systems that do not learn explicit rules but rather stochastic structures such as hidden Markov models (eg, [6]).

Following the standard methodology, an adaptive IE system is evaluated by invoking a set of learned rules on a disjoint set of test documents, and comparing the extracted templates to the reference templates TEMPS associated with the test data. The following notation indicates this methodology:

$$\text{INV}(\text{LEARN}(\text{TRAIN}), \text{TEST}) \rightarrow \text{TEMPS}$$

As demonstrated in Section 3, a direct application of conventional IE techniques to our multi-document extraction task yields poor performance. Our approach to multi-document extraction is two-fold. First, we pre-process the training data before it is submitted to the LEARN algorithm. Second, we post-process the fields extracted by INV before comparing them to the reference templates. Informally, we can characterize our multi-document extraction approach as follows:

$$\text{POST}(\text{INV}(\text{LEARN}(\text{PRE}(\text{TRAIN})), \text{TEST})) \rightarrow \text{TEMPS}$$

As described in the Sections 2.3 and 2.4, we have investigated several ways to PREprocess the training data as well as to POSTprocess the extracted content. An IE system can make two kinds of errors: false negatives (resulting in low recall) and false positives (resulting in low precision). Our pre-processing strategies are designed to improve recall, while our post-processing strategies attempt to improve precision.

2.3 Pre-processing the training set

Manually annotating training documents is expensive and error-prone, so a central challenge for adaptive IE is to be able to learn from small amounts of training data. Generally, insufficient training data results in an IE system that suffers from poor recall. This issue is particularly acute in the sort of specialized,

sparse multi-document extraction task we studied, in which the small community of users may not generate much training data even after months of email conversations, yet they still expect a robust IE system.

Several researchers have investigated so-called active learning strategies that ask humans to annotate only those documents that will actually lead to improved performance [5, 12, 13]. Our document preprocessing step is based on an alternative approach. Rather than assuming any control over the annotated documents, we have explored ways to augment the existing training data in an attempt to “trick” the learning algorithm to generalize more effectively.

In particular, the goal of our PREprocess strategy is to improve recall by automatically creating additional synthetic training documents from those provided by the human annotator. The expectation is that these additional documents will improve the recall of the learned extraction rules. (Of course, overall performance depends on both precision and recall. In Section 2.4 we discuss how the POSTprocess step will ensure adequate precision.)

As shown in Figure 2, we have explored three distinct PREprocessing strategies. The general idea is to make a “copy” of one of the original training documents, and then modify the copy in one of three ways:

1. **Replace.** Replace field values with alternative values mined from various field-specific Web sources. For example, we harvested a list of people’s names from the U.S. Census web site, which were used to replace fields that contain a person’s name.
2. **Scramble.** First convert each training document into the set of fragments that are *not* extracted. For example, a seminar announcement email containing a start time, speaker name and location, is converted into four segments: the text before the speaker, the text between the speaker and the start time, the text between the start time and the location, and the text after the location. Create the synthetic document by replacing the original document’s inter-field fragments with suitable fragments randomly selected from other training documents.

3. Replace & Scramble.

Make both of the above modifications.

The synthetic documents created by PREprocess are generally neither semantically meaningful nor grammatically correct. But from the perspective of the learning algorithm, we heuristically assume that they are annotated in the same way that a human would have annotated them if requested, and thus can serve as additional weak training data.

In particular, we expect that this process will increase recall. An important reason for poor recall is that the learned rules over-fit the training data. For example, a rule for identifying a seminar speaker might stipulate that the speaker’s name is “John”, since this is a common first name. By randomly inserting alternative names (“Henry”, “Archibald”, ...) we are encouraging the learning algorithm to generalize beyond the superficial cue “John”.

2.4 Post-processing extracted fields

Document PREprocessing is designed to improve recall. The later POSTprocessing step is designed to increase precision. The basic issue—like any situation in which recall and precision compete against each other—is that learned rules that extract lots of true positives are also likely to extract false positives.

In single-document extraction tasks, there are no constraints beyond what explicitly available in the document being processed. In contrast, multi-document extraction tasks offer the possibility of disambiguating one document on the basis on the other documents in its thread. As an extreme example, our email corpus (see Section 3) contains multiple copies of the same message, one in each thread to which it is relevant, and different fragments are supposed to be extracted depending on the context (ie, the thread in which it occurs). Clearly, it is impossible for any single-document IE system to achieve high performance in this case!

Our post-processing strategies revolve is to use knowledge learned about the temporal and/or structural inter-relationships between documents in a thread, in order to prune a large candidate set of extracted fragments to a smaller, more accurate set.

By doing so, recall remains relatively high while precision is increased.

In more detail, the POSTprocess step works as follows. The learned rules are first invoked on each of the N documents in a thread, to create our large candidate set, complete with extraction confidence values. We then start to fill the template using the best values from document 1. As we move through the documents in temporal order (2, 3, 4, ..., N), we replace values in the filled template only if the extraction confidence of a new slot fragment exceeds that of a fragment already extracted. Once we reach the the N ’th document, we have filled the template in the best possible way using temporal ordering.

This algorithm can be confused by, for example, a single high-confidence (but incorrect) fragment causing POSTprocess to discard several correct occurrences of the correct fragment that were extracted with lower confidence. In Section 5 we discuss our ongoing extensions and generalization of this simple approach to POSTprocessing.

3 Experiments

We have performed several experiments that demonstrate that our techniques are useful in the extraction of information from multi-document threads.

3.1 Postgraduate e-mail corpus

Our experiments are conducted on a collection of email conversations between prospective postgraduate students, their research supervisors, and the postgraduate director, of the Computer Science Department at University College Dublin. These conversations typically involve a student wishing to apply for a postgraduate position, and the director and supervisor collating the information necessary to process the application.

As mentioned in subsection 2.3, we expect datasets from these specialised domains to be especially sparse. Our dataset was obtained by sifting through 4 years of archived email messages, and yet in the end, it consisted of only 107 emails, which make up 48 distinct conversational threads.

Date: Mon, 30 Jul 2001 12:50:18 +0100
 From: ^{John Doe} <jdoe@ucd.ie>
 Subject: Re: postgrads
 To: John Smith <jsmith@ucd.ie>
 I left these 2 applications in your mailbox,
 as well as one from <sna>Jim Jones</sna>
 whos transferring from DCU to here on
 <sdt>Sept 1st</sdt>.
 regards, ^{John}

(a) Original Document

Date: Mon, 30 Jul 2001 12:50:18 +0100
 From: ^{Fred Everret} <jdoe@ucd.ie>
 Subject: Re: postgrads
 To: John Smith <jsmith@ucd.ie>
 I left these 2 applications in your mailbox,
 as well as one from <sna>Alan Green</sna>
 whos transferring from DCU to here on
 <sdt>October</sdt>.
 regards, ^{Fred Everret}

(b) Document after Strategy One (Replace)

From: dcuddy@ucd.ie
 To:^{John Doe} swatson@ucd.ie
 Subject: creedon
 Date: Tue, 15 Aug 2000 12:01:03 +0100
 for <sna>Jim Jones</sna>is definitely
 starting an MSc with me in <sdt>Sept
 1st</sdt> all supervised by both
 ^{John} will take care of
 officially registering your at this
 end.
 regards,
 -- Alex

(c) Document After Strategy 2 (Scramble)

From: dcuddy@ucd.ie
 To:^{Fred Everett} swatson@ucd.ie
 Subject: creedon
 Date: Tue, 15 Aug 2000 12:01:03 +0100
 for <sna>Alan Green</sna>is definitely
 starting an MSc with me in <sdt>October</sdt>
 all supervised by both ^{Fred Everett}
 will take care of officially registering your
 at this end.
 regards,
 -- Alex

(d) Document After Strategy 3 (Replace & Scramble)

Figure 2: An example document before and after the three PREprocess strategies.

The template is described in detail in Figure 3. Not all values for these slots are guaranteed to be present within a thread of emails. This document corpus is available to the research community; contact the authors for details.

3.2 Baseline performance

Our experiments are based on the (LP)² adaptive IE algorithm [4]. This algorithm learns rules that incorporate syntactic and contextual information to help locate the probable position of pieces of desired information in a new unseen document. These rules are used to insert tags into new unseen documents. When all the rules have been applied, pieces of text from the new document that are surrounded by correctly paired tags are deemed to be extracted. The algorithm also learns a second set of rules that can alter the position of tags already placed in order to make the encapsulated text better fit the slot it is

designated for.

As a baseline, Figure 4 shows the average cross-validated F1 performance of (LP)² on the email corpus. These data demonstrate that this task is inherently very challenging. Even when trained on most of the available data, F1 is well below 50% for most fields. F1 only exceeds 50% on one slot (**sup**). Unlike the others, this field takes values from a small finite set, and it appears that the learned rules simply memorize these values. Note that (LP)² is a state-of-the-art adaptive IE algorithm across a wide variety of domains, so we attribute these poor results to the task rather than the learning algorithm.

3.3 Pre-processing experiments

As described in Section 2.3, we have explored several ways to construct additional synthetic training examples from the manually-annotated training documents.

Slot	Meaning	Example
sna	Student name	John, Amy O’Neill should also
sno	Student ID number	his ID is 99-459-381 . He got a 1st
sup	Prospective supervisor	From: John Smith <john.smith@ucd.ie>
sdt	Expected start date	she should start in October . thanks
deg	Primary degree type	he got a B.sc. in Physics from
sub	Subject of primary degree	he got a 1st class B.sc. in Physics from
grd	Grade achieved in primary degree	BSc, 2002, 2.1 (Computer Science)
ins	Institution awarding primary degree	she finished at trinity in 99
dyr	Year of primary degree	BSc, 2002 , 2.1 (Computer Science)

Figure 3: The postgraduate email template comprises nine fields.

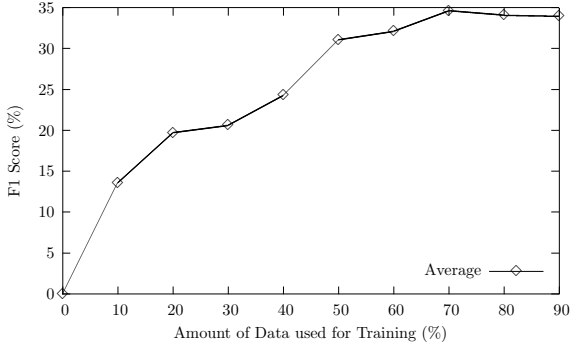


Figure 4: Baseline F1 performance of (LP)² on the email corpus.

Figure 5 shows the F1 performance as function of the numbers of new threads added to the training set, for each of the three strategies. For example, in “3” column, PREprocess generated three synthetic threads from each original thread.

These data demonstrate that using a modest number of synthetic training documents yields a dramatic improvement in F1. As the ratio between synthetic and original documents increases, performance declines, and eventually deteriorates below the baseline.

Based on these results, we fixed the number of synthetic threads at 5 per original thread, and adopt the **Replace** strategy. Figure 7 shows a learning curve for this configuration as well as the baseline with neither PRE nor POSTprocessing. (We discuss the two

Strategy	Synthetic threads per original				
	0	1	3	5	8
Replace	20.13	23.62	24.98	26.47	24.94
Scramble	20.13	21.45	23.02	24.63	22.51
Both	20.13	22	23.69	-	-

Figure 5: F1 performance for three PREprocessing strategies, as a function of the number of synthetic threads created.

POSTprocessing curves in Section 3.4.) We observe in this graph that PREprocess improves F1 by about 15% compared to the baseline.

Recall that the motivation of PREprocess is to increase recall. Figure 7 showed that F1 increases, and Figure 6 confirms that this was partly due to a large increase in recall. Fortunately, this did not come at the cost of decreased precision; indeed precision rose substantially as well.

Finally, we also tried our PREprocess techniques on Freitag’s seminar announcements dataset [7]. The results were quite different: performance deteriorated slightly (though less than 5%) rather than improved. We defer to future work an exploration of the conditions under which our approach is suitable.

3.4 Post-processing experiments

Returning to Figure 7, we also show the results of the POSTprocess algorithm describe in Section 2.4. As

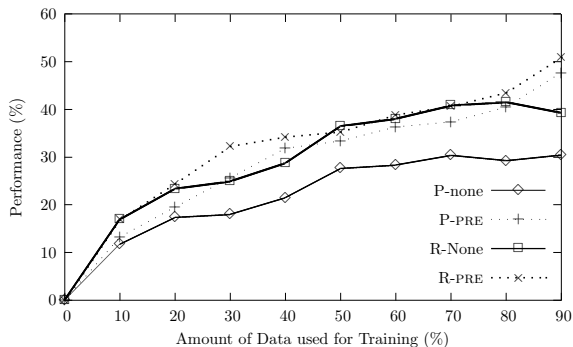


Figure 6: Effect of using PREprocessing on precision and recall.

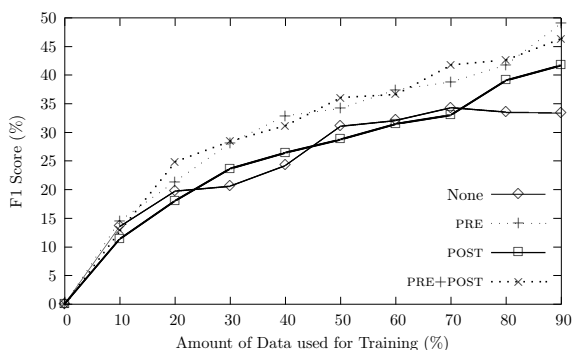


Figure 7: Performance with just PREprocessing, just POSTprocessing, both PRE and POSTprocessing, and neither.

with PREprocess, we see a substantial improvement over the baseline of approximately 8%. If we combine PRE and POSTprocessing, we see an improvement of about 13%.

It would appear that both PREprocess and POSTprocess contribute to improved performance, but we note that the performance gains are not cumulative. In Section 5, we describe improved POSTprocessing strategies that we hope yield further improvements.

4 Related Work

While there is much work in the field of information extraction, there is virtually no work on multi-document extraction. There has been considerable effort on the task of discourse processing, which involve recognizing discourse acts in speech segments. Discourse processing has been applied to IE [9], but was found to give disappointing results on the MUC-6 Business News task. However, Kehler’s experiments involved single-document rather than multi-document extraction, so no document interrelationships were assumed or used.

Another related sub-field of discourse processing is the the problem of dialogue act modeling [15], the analysis of conversational structure by identifying and labeling dialogue acts (such as “question” or “statement”). Dialogue act modeling is similar in spirit to our problem, in that we could view a document thread as a series of dialogue acts, and then try to identify the various pieces of information within them using the labeling of the dialogue acts as temporal and structural clues.

5 Discussion and Future Work

Information extraction is a powerful approach knowledge management, but existing techniques emphasize single-document extraction tasks in which templates do not span across documents. However, many workflow scenarios (such as intelligent email management) involve information distributed across several documents, so novel approaches to multi-document extraction are required.

We described a two-phase approach to multi-document extraction. In a pre-processing phase, the training data is augmented with synthetic documents in an attempt to improve recall. In a post-processing phase, the temporal relationships between documents are exploited to disambiguate extraction decisions. Our approach resulted in a 15% improvement in a challenging real-world multi-document extraction task.

Our current work is focused on enhancing our POSTprocessing strategies. In the extraction results,

we often encounter multiple results being extracted for a single field (i.e. "John Smith" and "Dr John Smith" being extracted as candidates for a supervisors name). We are experimenting with the use of string similarity metrics, such as edit distances, to merge fragments that are deemed to be sufficiently similar before pruning of the candidate set begins, and in doing so raising their confidence values. This should help to remove ambiguous results as well as boosting the score of fragments that are extracted multiple times. However, this technique has so far yielded mixed results.

A more sophisticated strategy that we intend to investigate would be to learn temporal regularities from document threads, in order to disambiguate between several possible extraction values. For example, if document number n in a thread contains a request for a student number ("Can you please tell me your student number?"), then document number $n + 1$ is likely to contain a student number. This could be accomplished by mining the thread tag sequences for frequent temporal patterns [1], and then adjusting the confidence of extracted values to reflect this additional inter-document evidence.

Acknowledgments. This research was supported by grants SFI/01/F.1/C015 from Science Foundation Ireland, and N00014-03-1-0274 from the US Office of Naval Research. We thank Fabio Ciravegna for access to (LP)².

References

- [1] R. Agrawal and R. Srikant: Mining Sequential Patterns. *Proc. 11th International Conf. on Data Engineering* (1995) 3–14
- [2] J. Cadiz and L. Dabbish and A. Gupta and G. Venolia: Supporting email workflow. *Microsoft Research Technical Report*, (2001) MSR-TR-2001-88.
- [3] M. E. Califf and R. J. Mooney: Relational Learning of Pattern-Match Rules for Information Extraction. *Working Notes of AAAI Spring Symposium on Applying Machine Learning to Discourse Processing* (1998) 6–11
- [4] F. Ciravegna: (LP)², an Adaptive Algorithm for Information Extraction from Web-related Texts. *Proc. IJCAI-01 Workshop on Adaptive Text Extraction and Mining* (2001)
- [5] F. Ciravegna et al. User-System Cooperation in Document Annotation based on Information Extraction *Proc. 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)* (2002)
- [6] D. Freitag and A. McCallum: Information Extraction with HMMs and Shrinkage. *Proc. AAAI-99 Workshop on Machine Learning for Information Extraction* (1999)
- [7] D. Freitag: Machine learning for information extraction in informal domains. *Ph.D. Dissertation*, Carnegie Mellon University (1998)
- [8] D. Freitag and N. Kushmerick: Boosted Wrapper Induction. *Proc. 17th National Conference on Artificial Intelligence* (2000) 577–583
- [9] A. Kehler: Learning Embedded Discourse Mechanisms for Information Extraction. *Proc. AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*. (1998)
- [10] N. Kushmerick: Wrapper induction: Efficiency and Expressiveness *Artificial Intelligence* (2000) **118**(1–2):15–68.
- [11] I. Muslea and S. Minton and C. Knoblock: Hierarchical Wrapper Induction for Semistructured Information Sources. *Autonomous Agents and Multi-Agent Systems* **4** **1/2** (2001) 93–114
- [12] I. Muslea and S. Minton and C. Knoblock: Active learning with strong and weak views: A case study on wrapper induction *Proc. 18th International Joint Conference on Artificial Intelligence* (2003)
- [13] T. Scheffer and S. Wrobel Active Learning of Partially Hidden Markov Models *Proc. ECML/PKDD Workshop on Instance Selection* (2001)
- [14] S. Soderland: Learning Text Analysis Rules for Domain-specific Natural Language Processing. *PhD thesis*, University of Massachusetts (1996)
- [15] A. Stolcke et al. : Dialogue Act Modelling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics* **26**(3) 339–373 (2000)

An Analysis of Ontology-based Query Expansion Strategies

Roberto Navigli and Paola Velardi
Dipartimento di Informatica
Università di Roma “La Sapienza”
{navigli,velardi}@di.uniroma1.it

Abstract

Sense based query expansion never proved its effectiveness except for the so-called “open domain question answering” task. The present work is still inconclusive at this regard, due to some experimental limitations, but we provide interesting evidence suggesting new guidelines for future research. Word sense disambiguation is in fact only one of the problems involved with sense based query expansion. The second is how to use sense information (and ontologies in general) to expand the query. We show that expanding with synonyms or hyperonyms has a limited effect on web information retrieval performance, while other types of semantic information derivable from an ontology are much more effective at improving search results.

1 Introduction

Despite the growing effort of the Semantic Web community to demonstrate that ontologies, and, in general, semantic knowledge may indeed improve the accessibility of web documents by humans and machines, no strong experimental results are yet available to support this convincement. The most important Information Retrieval (IR) conferences (SIGIR, TREC)¹ show the predominance of standard keyword-based techniques, improved through the use of additional mechanisms such as document structure analysis (Cutker et al. 1999) and query expansion using statistical methods (Carpineto et al. 2002) and query logs (Cui et al 2002). High-performant search engines rely also on the exploitation of the hypertextual relations in a document, using anchor analysis (Eiron and McCuley 2003) and link analysis (Borodin et al 2001). The effectiveness of the various techniques depends on the task, e.g. subject finding vs. site retrieval, as well as on the dimension of the query (short vs. long). Query expansion seems

particularly useful for short queries of two-three words, that represent the standard case for users of web search engines.

Except for the so-called “open domain question answering” task (Moldovan et al. 2002), the use of knowledge bases in state-of-art web retrieval systems is almost absent. Published results on sense based query expansions are not very recent (Voorhees, 1993) (Sanderson, 1994). A more recent work (Gonzalo et al., 1998) analyzes the effect of expanding a query with WordNet synsets, in a “canned” experiment where all words are manually disambiguated. Gonzalo and his colleagues show that a substantial increase in performance is obtained only with less than 10% errors in the word sense disambiguation (WSD) task. Since WSD is known as one of the hardest problems in Artificial Intelligence, this study left us with little hope that sense based query expansion might indeed rival with statistical methods.

We believe that the complexity of WSD is only one of the problems, the second being *how to use sense information to effectively expand the query*. In the literature, sense based query expansion is performed replacing senses with taxonomic information, e.g. synonyms or hyperonyms. However, the most successful query expansion methods seem to suggest that the best way to expand a query is by adding words that often co-occur with the words of the query, i.e. words that, on a probabilistic ground, are believed to pertain to the same *semantic domain* (e.g. *car* and *driver*). Query expansion terms are extracted either from an initial set of top retrieved documents (Carpineto et al. 2002) or from query logs, i.e. associations between a query and the documents downloaded by the user (Cui et al 2002). This latter source of co-occurrence information is obviously more precise, but proprietary.

In this study, we experiment the possibility of using ontological information to extract the *semantic domain* of a word. Rather than using taxonomic relations for sense based query expansion (e.g. synonyms and hyperonyms) we expanded with the words in a sense *definition*. In our experiment, we use the Google retrieval engine, the

¹ <http://www.informatik.uni-trier.de/~ley/db/conf/sigir/>
<http://trec.nist.gov/>

search topics of the TREC 2001 web track² to query the web, and WordNet 1.6³ to extract word senses and sense-related information. Our results are preliminary, both because of the limited size of the experiment, and because of some limitations imposed by the use of Google. Still, we show a systematic improvement over the unexpanded query, especially when query expansion terms are chosen from the sense definitions of the query words. Interestingly, the improvement is considerable even when word sense disambiguation performance is lower than the 90% suggested by Gonzalo and his colleagues.

The paper is organized as follows: section 2 describes the disambiguation algorithm, which relies on the WordNet lexical knowledge base. Section 3 describes the different query expansion methods adopted in the experiment. Section 4 presents a discussion of the results. Conclusions are drawn in section 5.

2 Word Sense Disambiguation

Word sense disambiguation is known as one of the most complex tasks in the area of artificial intelligence. We do not even attempt here a survey of the field, but we refer the interested reader to the Senseval home page (<http://www.senseval.org/>) for a collection of state of art sense disambiguation methods, and the results of public competitions in this area. During the most recent Senseval evaluation, the best system in the English all-words task (Mihalcea and Moldovan, 2001) reached a 69% precision and recall, a performance that (Gonzalo et al., 1998) claim to be well below the threshold that produces improvements in a text retrieval task.

However, for a query expansion task it is not necessary to pursue high recall, but rather high precision. As we show in sections 3 and 4, even expanding only monosemous words in a query may produce a significant improvement over the unexpanded query.

Therefore we developed an algorithm that may be tuned to produce high precision, possibly at the price of low recall. The algorithm belongs to the class of structural pattern recognition methods (Pavlidis, 1977). Structural pattern recognition is particularly useful when instances have an inherent, identifiable organization, which is not captured by feature vectors. In our work we use a graph representation to describe instances (word senses).

Shortly, the algorithm is as follows:

Let $Q = \{w_1, w_2, \dots, w_n\}$ be the initial query (stop words are pruned as usual)

Let

$$S(w_k) = \{S_j^k \mid S_j^k \in \text{Synset}_{\text{WordNet}}(w_k), w_k \in Q\}$$

² <http://trec.nist.gov/pubs/trec10/papers/web2001.ps.gz>

³ <http://www.cogsci.princeton.edu/~wn/>

be the WordNet synonym sets (synsets) of w_k , $k=1, \dots, n$. Let further

$$C_x = (S_{x_1}^1, S_{x_2}^2, \dots, S_{x_n}^n)$$

be a possible configuration of senses for Q (x_k is a sense index between 1 and the number of possible senses for w_k).

For each configuration C_x , do the following:

1. Create semantic networks for each sense;
2. Intersect semantic networks;
3. Assign a score the configuration.

Finally select $C_{\text{best}} = \arg \max_x (\text{Score}(C_x))$.

In the next sections the three steps will be described in detail.

2.1 Creation of semantic networks

For every $w_k \in Q$ and every synset S_j^k of w_k (where S_j^k is the j -th sense of w_k in WordNet) we create a *semantic net*.

Semantic nets are automatically built using the following semantic relations: *hyperonymy* (car *is-a* vehicle, denoted with $\rightarrow^@$), *hyponymy* (its inverse, \rightarrow^\sim), *meronymy* (room *has-a* wall, $\rightarrow^\#$), *holonymy* (its inverse, $\rightarrow^\%$), *pertainymy* (dental *pertains-to* tooth \rightarrow^\backslash), *attribute* (dry *value-of* wetness, \rightarrow^\equiv), *similarity* (beautiful *similar-to* pretty, $\rightarrow^\&$), *gloss* ($\rightarrow^{\text{gloss}}$), *topic* ($\rightarrow^{\text{topic}}$), *domain* (\rightarrow^{dl}).

Every relation is directly extracted from WordNet, except for *gloss*, *topic* and *domain*.

The *topic* and the *gloss* relations are obtained parsing with a NL processor respectively the SemCor⁴ sentences including a given synset S_j^k and WordNet concept definitions (called *glosses*). *SemCor* is an annotated corpus where each word in a sentence is assigned a sense selected from the WordNet sense inventory for that word; an example is the following:

Movement#7 itself was#7 the chief#1 and often#1 the only#1 attraction#4 of the primitive#1 movies#1 of the nineties#1.

The *topic* relations extracted from Semcor identify semantic co-occurrences between two related nodes of the semantic network (e.g. *chief#1* $\rightarrow^{\text{topic}}$ *attraction#4*).

As far as the gloss relation is concerned, it is worth noticing that words in glosses do not have sense tags in WordNet, therefore we use an algorithm for gloss disambiguation that is a variation of the WSD algorithm described in this section. For example, for sense #1 of

⁴ <http://engr.smu.edu/~rada/semcor/>

bus “a vehicle carrying many passengers; ...” the following relations are created:

$$\text{bus}\#1 \rightarrow^{\text{gloss}} \text{vehicle}\#1, \text{bus}\#1 \rightarrow^{\text{gloss}} \text{passenger}\#1$$

Finally, the *domain* relation is extracted from the set of domain labels (e.g. *tourism*, *chemistry*, *economy*...) assigned to WordNet synsets by a semiautomatic methodology described in (Magnini and Cavaglia, 2000). To reduce the dimension of a SN, we consider only concepts at a distance not greater than 3 relations from S_j^k (the SN center). The dimension of the SN has been experimentally tuned.

Figure 1 is an example of SN generated for sense #1 of *bus*.

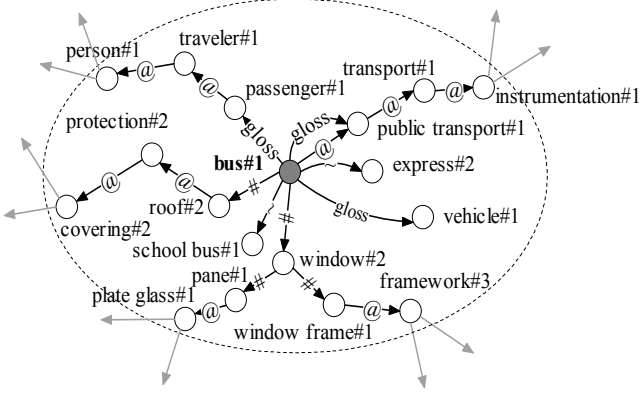


Figure 1. The semantic net for sense 1 of *bus*.

2.2 Intersecting semantic networks and scoring configurations

Let then $SN(S_j^k)$ be the semantic network for sense j of word w_k . Given a configuration of senses C_x , for a query Q , semantic networks are intersected pair-wise, and the number of common nodes are counted. Let $SN(S_j^k) \cap SN(S_m^l)$ be one such intersection. Common nodes S are those that can be reached from both SN centers through directed paths, e.g.: $S_j^k \rightarrow^* S^* \leftarrow^* S_m^l$ where \rightarrow^* denotes a sequence of nodes and arcs of any type.

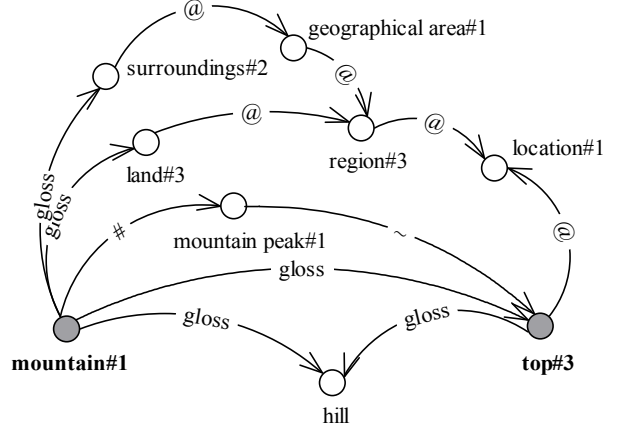


Figure 2. The patterns between *mountain#1* and *top#3*.

Figure 2 shows an example of intersection between the SN of sense 1 of *mountain* and the SN of sense 3 of *top*. There are 2 common nodes (*location#1* and *hill#1*), plus the direct gloss relation between the two central senses (therefore also the SN center *top#3* is common, according to our definition).

For each sense configuration, the score is computed as the total number of common nodes (e.g. 3 in the previous example):

$$\text{Score}(C_x) = \sum_{S', S'' \in C_x: S' \neq S''} |SN(S') \cap SN(S'')|$$

Furthermore, common nodes are ordered according to the inverse of the length of intersecting paths in which they participate. Let then $[S]^k$ be the ordered list of shared nodes in a configuration C_x .

3. The experiment

The objective of the experiment described in this paper is only in part the evaluation of the WSD algorithm described in previous section, that is still under refinement. Rather, our purpose is to obtain a better insight on the *use* of sense information for improving web search.

To this end, we used five sense-based expansion methods, and two strategies to choose expandable words. The following expansion methods are explored:

1. **Synset** expansion: “expandable” words are replaced by their synsets, retrieved by the algorithm of previous section.
2. **Hyperonym** expansion: “expandable” words are augmented by their WN direct hyperonyms.
3. **Gloss synset** expansion: “expandable” words are augmented by the synsets of its glosses

(disambiguated by an ad-hoc version of our WSD algorithm).

4. Gloss words expansion: “expansible” words are augmented with the words in their glosses.
5. Common nodes expansion: “expansible” words are augmented with the words whose synsets are in $\{S_j\}^x$.

According to the first strategy, expansible words are only monosemous words. In the second, we expand words whose synset, selected according to the WSD algorithm of section 2, has at least k ($k > 0$) nodes in common with other synsets of the query. The first strategy ensures maximum sense disambiguation precision, while the second allows it to tune the best precision-recall trade off, through the parameter k .

We queried the web with the first 24 of the 50 queries used in the TREC2001 web track. The queries (called “topics”) include the actual query (*title*) but also text to explain the query (*description*) and describe precisely the type of documents that should be considered relevant (*narrative*).

For example:

```
<top>
<num> Number: 518
<title> how we use statistics to aid our decision making?
<desc> Description:
Find documents that reference the use of statistical
data in decision-making.
<narr> Narrative:
A relevant document will describe a specific statistical
method that is used to assist decision-making.
</top>
```

Clearly, *description* and *narrative* texts cannot be used to expand the query, but only to manually verify the correctness of retrieved documents, as we did. To query the web, we used Google, which revealed not to be the best choice to exploit our algorithm, due to the limitation of 10 words per query. Therefore, for longer queries we are faced with the problem of choosing only a fragment of the candidate expansion words.

However, we felt that our results could be stronger if we show an improvement in performance using the most popular search engine.

For each query, we retrieved the first 10 top ranked pages without query expansion, and then we repeated the search with each of the sense based expansion methods outlined above. When expansion terms are synsets, terms of a synset are put in OR. Whenever plain query terms + expansion terms exceed the threshold imposed by Google, we simply choose the first words of the list, a strategy that is optimized only for method 5, since the list $[S]^x$ is ordered according to the strength of the intersection of each synset S .

The results are shown in Table 1a and b.

4. Discussion

Table 1a shows the results of the first four different expansion methods, when only monosemous words are expanded. The method five is not tested, since in each query rarely more than one word is monosemous. Consequently, intersecting paths between synsets are found only for one or two queries, which makes the evaluation not relevant.

Expanding only monosemous words is a very conservative policy, still, Table 1a shows interesting results. Every strategy produces an improvement. In particular, expanding with gloss words produces a 26,83% improvement over the plain query words. Interesting enough, the increase in performance (or at least a non-decrease) is systematic. The only critical cases are those where the query includes a named entity (e.g. topic 527: “can you info on booker t. washington?”). Since, regrettably, we did not use in this experiment any additional tool for the treatment of named entities, and since in the “TREC topics” names are often not capitalized, “booker” is interpreted as the monosemous concept *booker, booking agent*, with obvious consequences on retrieval performance.

Table 1b is the same experiment as for in Table 1a, but now the policy is to expand all words whose synset has at least one node ($k=1$) in common with some other synset of the chosen configuration. The sense configuration for ambiguous words is chosen according to our WSD algorithm of section 2.

Table 1b shows the results of six query types (unexpanded plus our five methods). The WSD algorithm attempted to disambiguate 52 words included in the 24 queries. The precision was 82,97% (39/47) and the recall 90,38% (47/52). This is a good result, though not comparable with WSD literature, given the limited size of the experiment. In agreement with the high precision requirement in (Gonzalo et al. 2000), the WSD precision is not enough to systematically improve over the monosemous words experiment, however the results are very close, in some case slightly better (synonyms) in some case slightly worst (gloss words). The gloss words expansion strategy achieves an improvement that is still very high, confirming that a better expansion strategy may overcome the problem of imprecise WSD.

This is a very interesting result, and even though the size of the experiment should be increased, this behavior seems very consistent across the various queries. Only in few cases some gloss word causes a decrease in performance. One interesting case is the topic “uniforms in public schools”. The gloss for *public school#1* is “a free school supported by taxes and controlled by a school board”. Here the word *tax* causes unwanted hits during web search. We would expect this problem be reduced by the “common nodes expansion” technique (the last column in Table 1b), but unfortunately this technique works very badly. In 11 cases no common nodes were

found⁵, and in the other cases the results are mixed, leading eventually to a decrease in performance with respect to the plain query.

Looking at the data, it appears that often there are interesting common nodes, but our node weighting method, though intuitive, does not capture them.

Furthermore, many common nodes are related to the query word synsets by a hyperonymy relation, a relation that was already confirmed as a bad expansion strategy. Same examples are useful here.

First, we provide an example of retrieved sense-based expansion words for the five methods, relative to the TREC topic 501: “deduction and induction in English?”

The first two lines show respectively, the senses chosen for each word by the WSD algorithm, and by the manual annotators (in some case more than one sense seemed appropriate). Then, the WordNet concepts extracted for each of the expansion policies are shown.

WSD: {deduction#3, induction#3, English#1}
WSD manual: {deduction#4{3, induction#3, English#1}
synsets: {English}, {generalization, induction, inductive_reasoning}, {deduction, entailment, implication}
hyperonyms: {reasoning, logical_thinking, abstract_thought}, {inference, illation}
gloss synsets: {England}, {detailed, elaborate, elaborated}
gloss words: relating, culture, England, characteristic, detailed, facts, general, principles, reasoning, implied, deduced, is_inferred, entailed
common nodes (with weights): {reasoning, logical_thinking, abstract_thought}:0.33, {syllogism}:0.16, {argumentation, local_argument, line_of_reasoning, line}:0.2, {thinking, thought, cerebration, intellection, mentation}:0.2, {deduction, deductive_reasoning, synthesis}:0.2, {analysis, analytic_thinking}:0.2, {conjecture}:0.2, {reasoning, logical_thinking, abstract_thought}:0.33

It is interesting also to provide examples of common patterns between semantic networks of word senses. In the previous query, several common patterns are found between *deduction#3* and *induction#3* (figure 3).

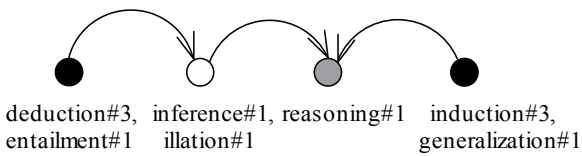


Figure 3. A path between *deduction#3* and *induction#3*.

where $[S_i]$ is the common node according to our definition.

Induction and *deduction* are semantically close words, and in fact the common node technique work well here. Often common paths are found also between less intuitively related words, e.g. in TREC topics “information

about what manatees eat” and “hair loss is a symptom of what diseases”. In the first query, the common nodes technique find patterns like that in figure 4.

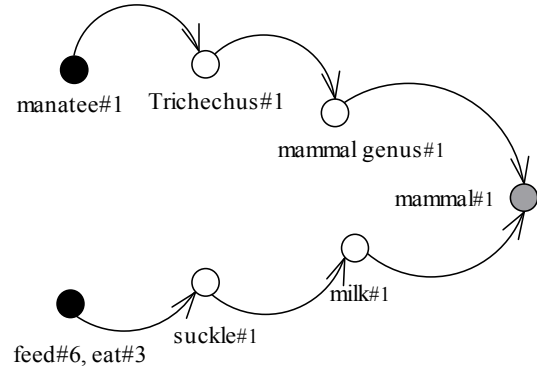


Figure 4. A path between *manatee#1* and *eat#3*.

The final set of chosen common nodes for this query is: *animal*: 0.25, *mammal*:0.2, *placental*:0.16, *animal_order*:0.16. These nodes are overly general, and cause noise during expansion. As shown in Table 1b, the number of relevant pages drops from 9 (plain query words) to 6 when using the common nodes technique for this query.

For the second query, the final set of common nodes is: *medical*:0.33 and *patient*:0.33. These nodes have the same level of generality as the query words. Contrary to “manatee eat”, common nodes improve “hair loss” search from 7 to 9 relevant pages.

In many cases, named entities are the cause of problems, since their related synsets are often not useful to expand the query (figure 5 shows such a common path for the topic history on cambodia?).

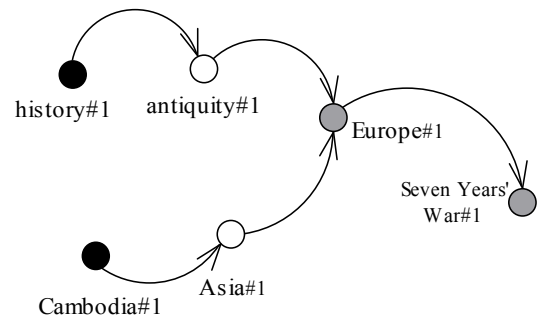


Figure 5. A path connecting *history#1* and *Cambodia#1*.

The problem here is that named entities are *instances*, not concepts, and should not be expanded at all. In WordNet, instead, there is an ontological confusion between instances and concepts, which are treated in the same way. As we already remarked, we plan to have an ad-hoc treatment of named entities in our future experiments.

⁵ in these cases (marked with *) the plain query word strategy is used, to allow a comparison with the other columns of Table 1b.

To summarize, each of the retrieved sense-based information types is in principle useful for query expansion, and even expanding only monosemous words in a query may provide a significant improvement in retrieval. Ideally, the sense-based expansion algorithm should be able to exploit and combine at best each of the available strategies, but this is matter of future research. Analyzing in detail the data, we found that words in the same *semantic domain* (and *same level of generality*) of the query words appear as the best candidates for expansion. Expanding with gloss words gives the best results by large, while hyperonyms expansion is a less performant strategy. The common nodes technique produces improvement (almost) only when the selected common nodes are related through non taxonomic relations (e.g. the *medical care* example), while it works quite badly if the selected common nodes are hyperonyms of the query word senses. These findings seem very consistent throughout our data, however, in order to declare our results conclusive, it is indeed necessary to experiment on much larger corpora, for example, over the TREC web corpus, an experiment that we plan to conduct in the near future.

5. Conclusions

In this paper we presented a word sense disambiguation method based on structured pattern recognition, and we used this method to explore several sense-based strategies for web search query expansion. By means of a small but interesting set of experiments, we could draw interesting conclusions on the type of sense-related information that appears more useful for web search, and obtain evidence on possible developments of the work.

Acknowledgments

This work has been in part funded by the MURST-CNR Web Learning national project.

References

- (Borodin et al 2001) Borodin a., Roberts G., Rosenthal J., Tsaparas P. : Finding Authorities and Hubs from Link Structures on the WWW" WWW10, may 1-5. 2001 Hong-Kong.
- (Carpineto et al. 2002) Claudio Carpineto, Giovanni Romano, Vittorio Giannini: Improving retrieval feedback with multiple term-ranking function combination. TOIS 20(3): 259-290 (2002).
- (Craswell and Hawking, 2002) Overview of the TREC-2002 Web Track, http://trec.nist.gov/pubs/trec11/t11_proceedings.html
- (Cui et al. 2002) Cui H. Wen J. Nie J. Ma W. Probabilistic Query expansion using query logs" WWW202, may 7-11, Hawaii, USA, ACM 1-58113-449-5/02/0005.
- (Cutler et al. 1999) Cutler M. Deng H. Maniccom S. Meng W. "A new study on using HTML structure to improve retrieval" 11th IEEE Conf. on Tools with AI, 1999.
- (Eiron and McCuley 2003) Eiron N. and McCuley K. Analysis of Anchor Text for Web Search, SIGIR 2003, Toronto, Canada.
- (Gonzalo et al. 1998) Julio Gonzalo Felisa Verdejo Irina Chugur Juan Cigarr'an "Indexing with WordNet synsets can improve text retrieval" Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP.
- (Magnini and Cavaglia, 2000) Magnini, B. and Cavaglia, G.: Integrating Subject Field Codes into WordNet. Proceedings of the 2nd International Conference on Language resources and Evaluation, LREC2000, Atenas.
- (Mahesh et al, 1999) Mahesh K.,Kud J. Dixon P. Oracle at TREC 8: A lexical Approach , Proc. of TREC 8, NIST, 1999.
- (Mihalcea and Moldovan, 2001) Rada Mihalcea, Dan I. Moldovan: A Highly Accurate Bootstrapping Algorithm for Word Sense Disambiguation. International Journal on Artificial Intelligence Tools 10(1-2): 5-21 (2001).
- (Moldovan et al. 2002) Moldovan, D. Harabagiu S., Girju R., Morarescu P., Llacatusu F., Novischi A., Badulescu A., Bolohan O.: LCC Tools for Question Answering, http://trec.nist.gov/pubs/trec11/t11_proceedings.html
- (Pavlidis 1977) T. Pavlidis Structured Pattern Recognition, Springer-Verlag Berlin 1977,
- (Sanderson, 1994) Sanderson M. Word Sense Disambiguation and Information Retrieval 17th Int. Conf. on Research and Development in Information Retrieval, 1994.
- (Voorhees 1993) Voorhees E. Using WordNet to disambiguate Word Senses for Text retrieval, ACM-SIGIR Pittsburgh, PA, 1993.

Table 1 a) Retrieved correct pages when using sense information only from monosemous words

Monosemous words only	Plain query words	+ synsets	+hyperonyms	+gloss hyperonyms	+ gloss words
Deduction induction	5	5	5	4	6
Prime factor	4	4	5	3	9
Vikings Scotland	4	4	7	9	7
Manatee eat	9	10	10	9	9
History skateboarding	7	7	7	9	9
Hair loss	7	7	9	4	8
Oppenheimer	3	3	6	9	9
Diseases smoking	9	9	10	9	8
Tornadoes	7	7	5	8	8
Earthquakes	9	9	7	4	10
Bell	5	5	5	7	9
Halloween	5	4	0	4	8
Titanic	1	1	1	3	3
Decision making	2	3	2	0	6
Black plague	4	1	4	5	5
Mojave	4	4	4	5	7
Booker Washington	3	3	1	0	0
Hygrometer	2	5	3	5	3
Cambodia	8	8	8	5	6
Hypnosis	7	7	10	8	8
School uniforms	7	7	10	8	8
Artists 1700	1	1	0	1	3
Canadian b. Codes	9	9	8	6	7
FHA	1	3	2	0	2
Avg. correct pages (over first 10)	5,125	5,25	5,208333	5,291667	6,5
% variation with respect plain query words		2,44%	1,63%	3,25%	26,83%

Table 1 b) Retrieved correct pages when using sense information for all disambiguated words

WSD with k=1	Plain Query words	+Synonyms	+Hyperonyms	+Gloss synsets	+Gloss words	+ Common nodes
Deduction induction	5	6	7	2	6	6
Prime factor	4	4	5	3	9	*4
Vikings Scotland	4	4	7	9	10	5
Manatee eat	9	10	10	10	10	6
History skateboarding	7	7	7	9	9	*7
Hair loss	7	7	8	6	5	9
Oppenheimer	3	3	6	9	9	*3
Diseases smoking	9	10	9	8	7	7
Tornadoes	7	8	5	5	7	2
Earthquakes	9	9	7	4	10	*9
Bell	5	5	5	7	9	*5
Halloween	5	4	0	4	8	*5
Titanic	1	1	1	3	3	*1
Decision making	2	1	3	4	5	2
Black plague	4	2	4	5	6	*4
Mojave	4	4	1	3	3	3
Booker Washington	3	3	2	6	3	1
Hygrometer	2	5	3	4	5	4
Cambodia	8	7	9	7	6	2
Hypnosis	7	7	10	8	8	*7
School uniforms	7	7	6	4	3	3
Artists 1700	1	1	0	1	2	*1
Canadian b. Codes	9	9	8	6	6	*9
FHA	1	3	2	0	2	3
Avg. correct pages over first 10	5,125	5,291667	5,125	5,291667	6,291667	4,5
%Variation with respect to plain query words		3,25%	1,63%	3,25%	22,76%	-12,20%

Combining Ontological Knowledge and Wrapper Induction techniques into an e-retail System¹

Maria Teresa Pazienza, Armando Stellato and Michele Vindigni

Department of Computer Science, Systems and Management, University of Roma Tor Vergata, Italy
{pazienza, stellato, vindigni}@info.uniroma2.it

Abstract. E-commerce and the continuous growth of the WWW has seen the rising of a new generation of e-retail sites. A number of commercial agent-based systems has been developed to help Internet shoppers decide what to buy and where to buy it from. In such systems, ontologies play a crucial role in supporting the exchange of business data, as they provide a formal vocabulary for the information and unify different views of a domain in a shared and safe cognitive approach. In CROSSMARC (a European research project supporting development of an agent-based multilingual/multi-domain system for information extraction (IE) from web pages), a knowledge based approach has been combined with machine learning techniques (in particular, wrapper induction based components) in order to design a robust system for extracting information from relevant web sites. In the ever-changing Web framework this hybrid approach supports adaptivity to new emerging concepts and a certain degree of independence from the specific web-sites considered in the training phase.

1 Introduction

The continuous growth of the Web accesses and e-commerce transactions is producing a new generation of sites: e-retail portals, willing to help end-users in choosing among similar products from different manufacturers, shown in an uniform context (to make easier their comparison). A number of commercial systems are being developed to automatically extract, summarize and show to the end-user relevant data from on-line product descriptions. Most of them neither use natural language technologies nor employ machine learning techniques, being based on shallow approaches that rely on page structure and/or HTML tags to retrieve information of interest. As a consequence, they must be manually tuned on specific pages of monitored sites, and do several assumptions, for example product names, prices, and other features to always appear in a fixed (or at least regular) order, or even pages to be expressed in uniform and monolingual manner (usually English) while it is generally not the case.

Extracting semi structured data from e-retail sites (and in general from the Web) appears to be a complex task, as target information is organized to be appealing and readable by human end-users and not by automatic extraction systems.

Ontologies play a crucial role in supporting information extraction, as they may be considered a formal vocabulary for the information and unify different views of a domain in a safe cognitive approach [9].

We describe here our contribution in building the knowledge base and the IE component as it has been developed inside CROSSMARC, an e-retail product comparison agent system (currently

¹ This work has been partially founded under the CROSSMARC project (IST 2000-25366) of the Information Society Technologies Programme of the European Union.

under development as part of an EU-funded project), where wrapper induction techniques ([1], [2], [4]), are boosted by background knowledge and linguistic analysis both to extend their capabilities and to further ease adaptation to domain changes.

CROSSMARC aims both to stress commercial-strength technologies based on language processing methodologies for information extraction from web pages and to provide automated techniques for an easy customization to new product domains and languages. Its technology currently operates for English, Greek, French, and Italian languages and is being applied to two different product domains: computer goods and job offers: the first one being characterized by brief and semi-structured descriptions, rich of technical terms and acronyms, while the second, IT job offers, contains wider linguistic descriptions. These domains have been chosen to evaluate the system in large on different presentation styles, contents, use of tables and layout aspects.

In the following section, an overall description of system architecture will be provided. Then we will focus on the Fact Extractor component, which exploits wrapper induction techniques to induce extraction rules on web pages semantically analyzed by other components. Special attention is on linguistic features.

2 Crossmarc Architecture

The overall CROSSMARC architecture [8] (see below Fig. 1) may be sketched as a pool of agents communicating via a dedicated XML language. Agents roles in the architecture are primarily related to three main tasks:

1. Process users' queries, perform user modeling, access the database and supply the user with product information
2. Extract information from the WEB: several processing steps are coordinated to find, retrieve, analyze and extract information from the Internet.
3. Store the extracted information in a database, in order to feed the system with the data to be later shown to the user

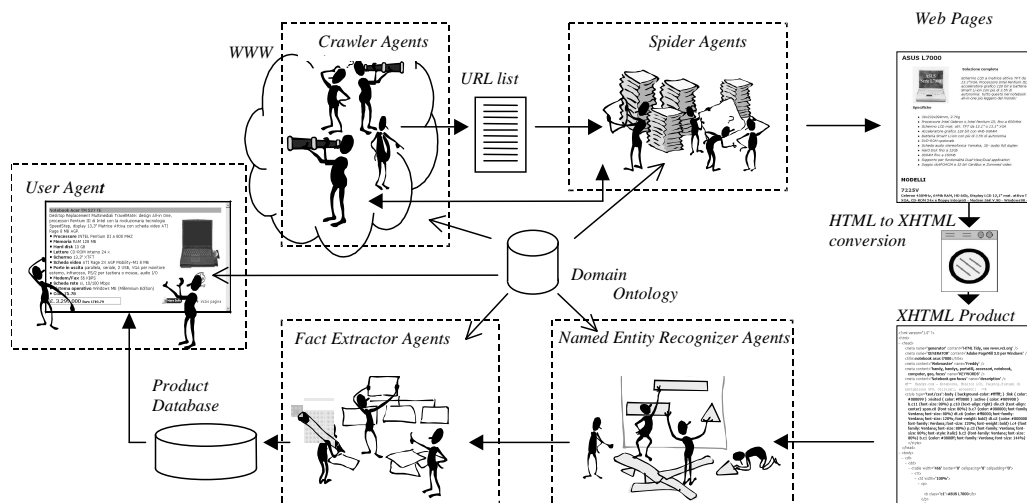


Fig. 1. Overall Crossmarc Architecture

Extraction agents can be divided into two broad categories, depending on their specific tasks:

- **Information retrieval agents (IR)**, which identify domain-relevant Web Sites (*focused crawling*) and return web pages inside these sites (*web spidering*) that are likely to contain the desired information;
- **Information Extraction (IE)** agents (one for each language) that process the retrieved web pages. There are specific roles for each step of the extraction process: a) *Named Entity Recognition and Classification* (NERC), i.e. recognition of concepts pertaining the domain of interest inside the web pages, b) identification of the number of products and their distribution in the web pages (*products demarcation*), c) *Fact Extraction* (FE), that is the extraction of products characteristics; all the information gathered during previous processing steps is merged in a XML data structure according to a common XML schema (the *Fact Extraction Schema*). Such a schema plays a pivotal role both in supporting interpretation of FE results by the product database feeder, and in providing consistency checking of the results.

Each agent commits to a shared ontology that drives its analysis throughout all the phases.

First of all, Named Entity Recognition and Classification (NERC) linguistic processors identify relevant entities in textual descriptions and categorize them according to the ontology [7], then inside the Fact Extraction and Normalization phase, these analyzed entities are aggregated to build a structured description of the identified product by exploiting the ontology organization. This description is composed of a set of features whose values are normalized to their canonical representation (as described into the ontology) for comparison purposes.

During the presentation of results to the end-user, correlations among different language lexicons and the ontology are exploited to adapt heterogeneous results to each language and locales.

The knowledge bases of the two domains (as well as lexicons for the four languages) [10], have been developed, accessed and maintained through a customized application based on Protégé-2000 API [5].

3 Wrapper Induction and Ontologies in the Crossmarc System

In the context of CROSSMARC, several FE components (one for language) have been developed by project partners. As a common characteristic, each Fact Extractor component implements wrapper induction techniques for extracting information pertaining to the products recognized inside each Web page. Boosted Wrapper Induction [4] has inspired the first version of the English Fact Extractor, STALKER [1] the Greek version of the Fact Extractor module, while the Italian one is a customized implementation of the Whisk algorithm [2].

In the following section more details on the Italian version of the Fact Extraction component and how it relies on semantic analysis carried on by other components of the CROSSMARC architecture will be provided.

3.1 CROSSMARC Italian Fact Extraction Component

WHISK [2] uses regular expressions as extraction patterns. It is not restricted to specific preprocessing of the text and hence it is good for structured, semi-structured and free texts. It induces a set of rules from hand-tagged training examples. WHISK rules are based on a form of regular expression patterns that identify both the context of relevant phrases and their delimiters for those phrases. Predefined domain-specific semantic classes are used, then applied to free text (the text is previously segmented into syntactic fields).

WHISK uses a covering algorithm inducing rules top-down, by first finding the most general one that covers the seed, then extending the rule by adding terms one at a time as long as it is below a certain threshold of error. Best performing rules are retained and the process is then reiterated until all the candidate extensions do not perform better than those produced in the previous step.

Although WHISK can learn either single or multi-slot rules, we considered the former ones: in fact target products may highly differ both in number and position of the features used for their description.

3.2 Boosting WHISK for CROSSMARC

The architecture of the Italian Fact Extractor (FE) component (see figure 2) consists of three different modules:

FE_Adapter: this module pre-processes the training set for the Learner and the Evaluator modules; it merges source web pages and annotation files into XML files, tokenizes the result (as WHISK works on tokenized input instances), and extracts from FE surrogate files a set of structured product descriptions, which will then be used by the training process to calculate the Laplacian Expected Error and to evaluate the output in terms of Precision and Recall statistics.

FE_Core: the FE core component. It performs both training and testing processes. These activities are implemented into a single module: in fact rules are continuously tested against the training set during the training phase, so training and testing need to be tightly coupled. If used in training mode, the output of this module is the set of induced rules; if used in testing mode, it produces a set of structured product descriptions, in the same format as the one produced by the FE_Adapter module (as they need to be compared during evaluation).

FE_Evaluator: it performs the evaluation of Whisk's product extractions on the test set by first identifying the number of correct extractions, then by computing and reporting statistics for precision and recall metrics.

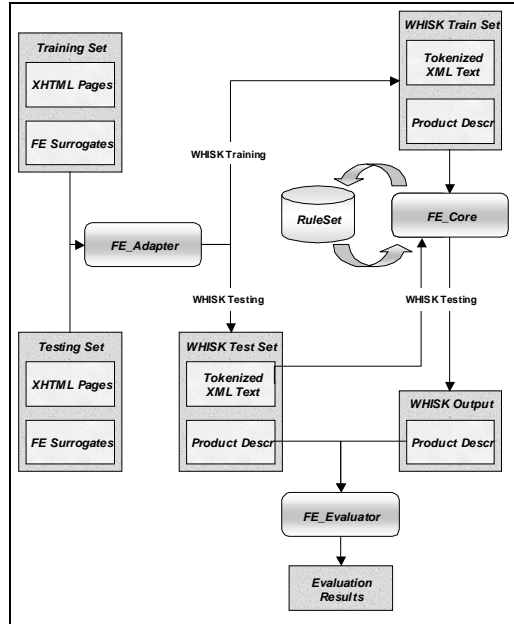


Fig 2: Interaction Diagram of Italian FE modules

The original algorithm has been customized to meet the specific needs of the CROSSMARC environment, through the following aspects:

a) Ontology Lookup. WHISK introduces the notion of Semantic Class to factorize disjunctive sets of terms into equivalence classes. A Semantic Class appears in an induced rule and provides a sort of generalization for the words it subsumes. In case a more complex analysis is needed to catch less evident phenomena, Whisk demands to other components (e.g., a syntactic parser) the task of providing such information in form of Semantic Tags wrapping recognized concepts.

These tags well fit on CROSSMARC needs, since all the Named Entities recognized by the NERC component are semantically classified into some ontological category (e.g. MANUFACTURER, PROCESSOR, CAPACITY, etc. in the Laptop Domain), while sets of Semantic Classes are defined on lists of lexical entries factorized according to the ontology. NERC components add to each XHTML page they process annotations for the named entities (NE), numeric expressions (NUMEX), time expressions (TIMEX) and terms they recognize. The type of NE, NUMEX, TIMEX is added as an attribute to the corresponding annotation. Here is an example of a tag inserted by a NERC component:

```
<NUMEX TYPE="LENGTH" onto-ref="OA-d0e1569" onto-attr="OA-d0e1569">
14.1 "
</NUMEX>
```

In the above examples 14.1" has been found and recognized as the length, expressed in inches, of something presented in the page. The FE component exploits the NERC annotations in order to identify which of the NEs, NUMEXs, TIMEXs, TERMS fill a specific fact slot inside a product description (e.g. which of the NUMEXs of type MONEY is the laptop's PRICE etc...), according to the above mentioned XML FE Schema.

By considering NE categories as Semantic Classes allows the induction system to focus on relevant product characteristics, thus providing an higher level of interpretation together with a strong bias on the search space of the wrapper induction algorithm. As a result, inferred rules become more sensitive to semantic information with respect to instance specific delimiters, while enhancing their robustness towards heterogeneous data.

b) Limiting Search Space of Induction when adding terms. WHISK original algorithm was conceived to operate on fragments of the source material containing the information to be extracted, while all the FE components must operate on entire web pages. As the algorithm complexity increases with the number of considered features and with the dimension of instances, in [2] the search for terms to grow rules is limited to a window-size of k tokens around an extraction slot.

To find a good trade-off between accuracy of rules (the wider the windows, the larger is the space of induction), and the time needed to learn them, we adopted two different windows, depending on two defined parameters:

- A Token window size (T_SIZE)
- A Semantic window size (S_SIZE)

The following strategy has thus been implemented to allow for a more stable window for rule improvement:

1. A Token Window of T_SIZE size is created near the element to be extracted.
2. Tokens in the Token Window are converted to Semantic Elements (Semantic Classes or Tags)

S_SIZE elements (both Semantic classes and remaining tokens) are considered when adding terms to the WHISK rule expression

c) Laplacian Expected Error versus Precision: rule application strategy. The Laplacian Expected Error Rate (i.e. $(e+1)/(n+1)$ where e is the number of wrong extractions over n extractions), adopted by Soderland as performance metrics, has been preserved for evaluation of the partial rules created during rule expansion; it expresses a good trade-off between rule precision and recall. This measure has been used instead of the Precision to prevent Whisk from choosing a large set of very specific rules covering only very few cases, thus preferring a more lightweight and general-purpose ruleset. When dealing with system's test or online work a different criterion has been adopted, to consider actual precision of the rules and to prevent abuse of the less precise (but more general) ones. We thus applied the following strategy:

- Ruleset Construction
 1. Each rule is characterized by its type (the kind of fact that it extracts), its Laplacian Expected Error and its Precision.
 2. Rules from the ruleset are sorted by Precision
 3. A threshold is set on Precision: induced rules with lower Precision are discarded; a different threshold for the Laplacian Expected Error is then considered: while it is not determinant for system accuracy, it helps to remove too specialized rules in order to enhance system performances.
- Rule appliance

For each rule, consider single extraction as a “candidate extraction”; for each candidate extraction:

1. discard the extraction if another candidate (whichever type it belongs) exists in the same span of tokens, else proceed to the next step.
2. discard the extraction if another candidate from a rule of the same type exists for the same product, else, proceed to the next step.
3. confirm the candidate as a valid extraction.

4 Evaluation of the Italian FE Component

The testing corpus for both NERC and FE components has been annotated by following a well known methodology ([3], [6]): a gold standard test set has been produced after comparison and merging annotations made by two different domain experts. The Italian test set for instance, consisted in a corpus of 100 web pages coming from 50 Italian sites of Laptop Vendors. A similar number of pages has been chosen for other languages.

Among the domain independent characteristics researched for the four languages in the 1st domain, product description category seems to affect the feasibility and difficulty of the IE tasks. Named Entity Recognition and Classification, for instance, is performed only within laptop product descriptions: a page including computer goods offers further to laptops, is more challenging than a page that includes only laptop descriptions. In fact, in the first case NERC component is more likely to recognize and classify non relevant names and expressions rather than in the second one, which consists only of descriptions of the actual products to be identified. Italian laptop vendor sites showed strong preference for one laptop description per page (45% of the corpus) and several laptop descriptions appearing in different lines/rows of a page (40%) with smaller preference for several laptop and other product descriptions appearing in different lines/rows of a page (8%). 42% of the pages in the Testing

corpus come from sites that do not appear in the Training corpus. A specific evaluation of each FE component in all the 4 different languages has been carried on: in table 1 evaluation results for the first domain (laptop computer offers) are summarized in precision and recall figures for all considered features for our Fact Extractor.

Table 1. Evaluation results for the Laptop Computers Domain on 4 different languages

FEATURE	ENGLISH		FRENCH		GREEK		ITALIAN	
	PREC	REC	PREC	REC	PREC	REC	PREC	REC
MANUFACTURER	0.89	1	0.99	1	1	1	1	0.99
PROCESSOR.	0.99	1	1	1	1	1	0.99	1
OPERATING SYSTEM	0.78	0.98	0.82	0.94	0.92	0.98	0.78	0.99
PROCESSOR SPEED	0.86	0.99	0.95	0.98	0.85	1	0.95	0.98
PRICE	0.99	1	1	1	1	1	1	1
HD CAPACITY	0.99	0.94	0.94	0.80	0.96	0.96	1	0.88
RAM CAPACITY	0.82	0.97	0.95	0.94	0.90	0.80	0.96	0.89
SCREEN SIZE	0.85	0.98	0.70	0.99	0.95	0.98	0.92	0.99
MODEL NAME	0.99	1	1	0.99	1	1	0.99	1
BATTERY TYPE	1	0.86	0.97	0.63	0.97	0.76	1	0.5
SCREEN TYPE	0.82	0.98	0.81	0.96	0.99	1	0.86	0.99
WEIGHT	0.98	1	0.96	1	1	1	0.92	1
AVERAGE VALUES	0.91	0.97	0.93	0.94	0.96	0.96	0.95	0.90

A straightforward comparison with Soderland's experiments on Whisk algorithm is not fully trustworthy, as we dealt with totally different domains and exploited richer linguistic analysis (and domain knowledge provided by the ontology). Bringing linguistic analysis in early processing phases provides more semantic evidences for the rule induction system reducing the dependency from specific page structures.

This results in rules with an higher level of coverage without significant loss in precision. All of these considerations motivated our architectural choices.

Moreover, the evaluation has been conducted inducing rules from a corpus of web pages with an high degree of heterogeneity to stress how the learned rules are less biased from rigid structure of the examined documents. Table 1 shows black-box evaluation of the FE components, as they are evaluated over error-free input from the previous processing steps (NERC and Product Demarcation), and does not provide sensitivity measures over noisy data. (Overall results for the IE system will be released by the end of the project foreseen for next Autumn).

5 Conclusions

At the cross point between knowledge-intensive IE systems with high maintenance needs and low-demanding machine learning algorithms, we have explored the possibility of combining the two approaches, leaving to the first one the knowledge of the domain required for the semantic analysis of the text while relying on the latter for explicit extraction of needed information.

Nowadays, in fact, the current trend in IE is in moving away from the rule-based approach, which relies on hand-crafted lexical resources and grammar rules, towards machine learning techniques in order to achieve swifter adaptation to new domains and text types.

Basing on this assumption, CROSSMARC reduces high system maintenance costs, which are closely related to modifications of tightly coupled rules and extraction methods, leaving only to Domain Experts and Knowledge Engineers the task of updating the ontology and lexicons of the domains, whilst machine learning techniques like Wrapper Induction can induce proper rules which exploit this

knowledge, thus furthering rapid adaptation to both new domains and changes in their conceptualisations.

References

- [1] I. Muslea, S. Minton C. Knoblock "Hierarchical Wrapper Induction for Semistructured Sources". *Journal of Autonomous Agents and Multi-Agent Systems*. Vol. 4, pp. 93-114, 2001.
- [2] Soderland S. "Learning Information Extraction Rules for semi-structured and free text. Machine learning. Volume 34 (1/3) pp. 233-272, 1999.
- [3] Boisen, S., Crystal, M., Schwartz, R., Stone, R. and Wischedel, R. "Annotating Resources for Information Extraction" *LREC 2000* pp. 1211-1214
- [4] Kushmerick N., "Finite-State Approaches to Web Information Extraction". in M.T. Pazienza Editor: *Information Extraction in the Web Era: Natural Language Communication for Knowledge Acquisition and Intelligent Information Agents*, LNAI2700, Springer-Verlag.
- [5] N. F. Noy, R. W. Fergerson, & M. A. Musen. "The knowledge model of Protege-2000: Combining interoperability and flexibility". *2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000)*, Juan-les-Pins, France, . 2000.
- [6] MUC-7 (2001) http://www.itl.nist.gov/iad/894.02/related_projects/muc/
- [7] C. Grover, S. McDonald, D. Nic Gearailt, V. Karkaletsis, D. Farmakiotou, G. Samaritakis, G. Petasis, M.T. Pazienza, M. Vindigni, F. Vichot and F. Wolinski (2002): "Multilingual XML-Based Named Entity Recognition for E-Retail Domains". *3rd International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas, Spain
- [8] M.T. Pazienza, M. Vindigni, (2002) "Mining linguistic information into an e-retail system" *Third International Conference on Data Mining Methods and Databases for Engineering, Finance and Other Fields*, Bologna, Italy
- [9] M. T. Pazienza and M. Vindigni. "Language-based agent communication". *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02)*, Sigüenza, Spain.
- [10] M.T. Pazienza, A. Stellato, M. Vindigni, A. Valarakos, V. Karkaletsis (2003) "Ontology integration in a multilingual e-retail system" *HCI International 2003*, Crete, Greece.

Meta-learning beyond classification: A framework for information extraction from the Web

Georgios Sigletos^{1, 2}, Georgios Paliouras¹,
Constantine D. Spyropoulos¹, Takis Stamatopoulos²

¹ Institute of Informatics and Telecommunications, NCSR “Demokritos”,
P.O. BOX 60228, Aghia Paraskeyh, GR-153 10, Athens, Greece
{sigletos, paliourg, costass}@iit.demokritos.gr

² Department of Informatics and Telecommunications, University of Athens,
TYPA Buildings, Panepistimiopolis, Athens, Greece
{sigletos, takis}@di.uoa.gr

Abstract. This paper proposes a meta-learning framework in the context of information extraction from the Web. The proposed framework relies on learning a meta-level classifier, based on the output of base-level information extraction systems. Such systems are typically trained to *recognize* relevant information within documents, i.e., streams of lexical units, which differs significantly from the task of classifying feature vectors that is commonly assumed for meta-learning. The proposed framework was evaluated experimentally on the challenging task of training an information extraction system for multiple Web sites. Three well-known methods for training extraction systems were employed at the base level. A variety of classifiers were comparatively evaluated at the meta level. The extraction accuracy that was obtained demonstrated the effectiveness of the proposed framework of collaboration between base-level extraction systems and common classifiers at meta-level.

1 Introduction

One common *meta-learning* approach, known as *stacked generalization* [18], deals with the task of learning a meta-level module to combine the predictions of multiple base-level learners. Base learners are treated as “black boxes”, i.e., only their output predictions are used, without considering the details of their functionality. The meta-level module is expected to achieve performance superior to each of the base learners, on unseen data.

Current work in meta-learning of this type focuses on the *classification* problem, i.e. learn how to assign the correct class value to each one of a set of different events, where each event is described by a *vector* of predefined *features*. Various studies, e.g. [1] and [16], have investigated which classifiers and data representations, either at the base or the meta level, and under which strategies, can lead to better classification results over unseen events.

In this paper, we attempt to drive the meta-learning framework outside the common feature-vector representation, employed in classification tasks. Our motivation is the

information extraction (IE) task, which can be defined as the process of directly extracting relevant text fragments from collections of documents and filling the slots of a predefined template. In particular, information extraction from Web pages is a simplified version of the harder free-text information extraction examined by the Message Understanding Conferences [11]. Despite its simplicity, though, it has gained popularity in the past few years, due to the proliferation of online sources, and the need to recognize useful pieces of information inside the Web chaos.

IE can be formulated as a regular-expression matching process within a document that is modeled by a sequence of lexical units (tokens). Learning a classifier to perform this task is unnatural and as a result specialized systems, like STALKER [12] and SoftMealy [10], learn extraction rules in the form of special types of regular expressions. However, there is a small number of approaches, which enumerate the possible text fragments that can be found within a document and then model the task as a binary classification one [5], [6]. In this case, the task is to learn whether or not a candidate fragment fills some template-slot. There is a number of problems associated with this approach, such as the exponential number of candidate fragments and the disproportionately large number of “negative” events. Therefore, it is particularly desirable to design an alternative framework that will use common IE systems.

Thus, the main contribution of this paper is a novel meta-learning framework that removes the constraint of employing classifiers at the base level, accommodating IE systems that *recognize* relevant text instances within documents, rather than classifying text fragments. The prediction output of the base IE systems is appropriately transformed into vectors of features, to be used for training a common meta-level classifier. We have experimented with three algorithms at the base level: two deterministic (STALKER [12] and (LP)² [2]) and a stochastic finite-state approach (Hidden Markov Models (HMMs) [13]). Four classifiers were evaluated at the meta level.

Section 2 reviews some basic theory in meta-learning for classification tasks. Section 3 illustrates our proposed framework. Section 4 presents the experimental results. Finally, the basic conclusions of this work are presented in Section 5.

2 Building a meta classifier – Basic theory

Wolpert [18] introduced an approach for constructing ensembles of classifiers, known as *stacked generalization* or *stacking*. A *classifier ensemble*, consists of a set of n classifiers C_1, \dots, C_n , called *base-level* classifiers and a *meta-level* classifier C_{ML} that learns how to combine the predictions of the base-classifiers. The base-classifiers are generated by applying n different classification algorithms on a labeled dataset $L_B = \{(x_k, y_k)\}$, where x_k and y_k are the features and the class value for the k -th instance vector respectively. The individual predictions of the base-classifiers on a different labeled dataset L_M , are used to train the meta-classifier C_{ML} . The predictions of the base-classifiers on L_M are then transformed into a meta-level set of classification vectors. At runtime, C_{ML} combines the predictions $P_M(x) = \{P_i(x), i = 1 \dots n\}$ of the n base-classifiers on each new instance x , and decides upon the final class value $y(x)$. The

predictions of the base-classifiers on x are transformed into a single vector representation, which is then classified by C_{ML} .

3 Building a meta-classifier for information extraction

The majority of IE systems that use machine learning, e.g. [10], [12], represent the acquired knowledge in the form of finite-state automata (FSA) or stochastic FSA [13]. Thus, IE becomes a task of matching a set of regular expressions within each document. We further assume a *single-slot* approach to IE that deals with extracting instances of isolated *facts* (i.e. *extraction fields*), whereby a different automaton is induced for each fact. For example, in a Web page describing CS courses, one automaton has to be induced for extracting instances of the “course number” fact, while a different one is required for extracting instances of the “course Title” fact.

3.1 Preliminaries

Our goal is to incorporate single-slot IE systems, into a meta-learning framework and thus exploit the advantages provided by the meta-learning theory, aiming at higher extraction accuracy. We make the following assumptions:

1. Let D be the sequence of a document’s tokens, and $T_i(s_i, e_i)$ a fragment of that sequence, where s_i, e_i are the *start* and *end* token bounds respectively.
2. Let $E_B = \{E_k \mid k = 1 \dots n\}$ be the set of n single-slot IE systems, generated by n different learning algorithms.
3. Let $I_k = \{i_j : T_j \rightarrow \text{fact}_j\}$ be the set of instances extracted by E_k , and fact_j the predicted fact associated with the text fragment T_j .

3.2 The proposed framework

We suggest a novel framework for combining the IE systems at base-level with a common classifier at meta-level, which is graphically illustrated in Figure 1.

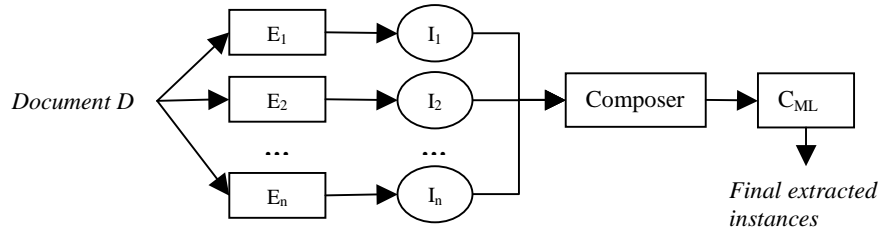


Fig. 1. Combining extraction systems and a common classifier at runtime

The starting point of the architecture depicted in Figure 1 is a *document D*, which is the input to each extraction system E_k , which generates a set of extracted instances I_k , over D . In contrast, the input to each classifier C_i in the common stacking framework, is an instance *vector* x , while the output is the predicted class value $P_i(x)$.

The combination of the base-level IE systems with the meta-level classifier C_{ML} depends on a *composer* module. At runtime, the input to the composer comprises the n sets of extracted instances I_1, \dots, I_n . The output of the composer must be a set of vectors, to be finally classified at meta-level by C_{ML} . Similarly in the training phase of C_{ML} , the output of the composer must be a set of *classified* vectors, based on information from the hand-labeled data. In order to translate the output of the IE systems to a fixed-length vector of events for C_{ML} , we make the following assumptions, affecting the functionality of the *composer* module:

1. Each event corresponds to a *distinct* text fragment $T(s, e)$ among all predicted instances in $\cup I_k$, $k = 1 \dots n$. Note that two text fragments $T_1(s_1, e_1)$ and $T_2(s_2, e_2)$ are different, if either $s_1 \neq s_2$ or $e_1 \neq e_2$.
2. The features of the new vector, associated with the text fragment T , are based on the predicted facts for T by the base-level IE systems. Note that for each distinct T among all instances in $\cup I_k$, $k = 1 \dots n$, there exists at least one instance $i_k : T \rightarrow \text{fact}_k$.
3. At runtime, each vector associated to a fragment T is to be classified into one of a set of nominal values, corresponding to the f different facts in the domain of interest, plus an additional value “false” if the text fragment is judged as not being an interesting fact.
4. During the training of C_{ML} , each vector associated to a T , is augmented with a class value, corresponding to the hand-labeled fact of the fragment. If T is not labeled, the new vector is assigned to the “false” class.

Consider the token table in Table 1(a), which is part of a page describing computer science courses. Table 1(b) shows the extracted instances by 2 base-level IE systems over the token table in Table 1(a). Note that the first system has not predicted a fact for the text fragment $T_2(27, 28)$.

Table 1. (a) Part of a token table for a page describing computer science courses. (b) Extracted instances by two base-level IE systems E_1, E_2 . (c) The distinct text fragments and the information associated to each T for constructing the new vectors

...	25	26	27	28	...
...	CS414	:	Operating	Systems	...

(a)

$T(s, e)$	E_k	$Fact$
$T_1(25, 25)$	1	Course Number
$T_1(25, 25)$	2	Course Number
$T_2(27, 28)$	2	Course Title

(b)

$T(s, e)$	Information for meta-level vectors
$T_1(25, 25)$	(1, Course Number), (2, Course Number)
$T_2(27, 28)$	(2, Course Title)

(c)

Table 1(c) shows the two *distinct* text fragments, each associated with a set of pairs $\langle E_k, \text{fact}_k \rangle$, where fact_k is the predicted fact by the k -th base-level IE system. The

information in those pairs will be used for building the two meta-level vectors – one for each distinct T .

3.3 Meta-level data representation

In this paper we experiment with two different vector representations:

1. *Numeric-feature* representation: each distinct text fragment T is modeled by a vector of f numeric features, each one corresponding to a fact of interest, e.g. *Course Number*, *Course Title*. During the training of the meta-classifier, the vector is augmented with an additional class feature, which is the true fact of T , according to the labeled document. For each fact predicted by an IE system, the corresponding feature value is incremented by one, starting from zero.
2. *Binary-feature* representation: each distinct text fragment T is modeled by a vector of $n*f$ binary features. The output of each of the n base-level IE systems is a set of f binary features. For each predicted fact, the corresponding feature is set to one. In case of ambiguous facts, more than one features will have the value one. All other features are set to zero.

The advantage of the first representation is that the number of features depends only on the number of the facts of interest, and remains fixed, independently of the IE systems employed at base-level. The advantage of the second representation is that the predicted facts of the base-level IE systems are modelled separately. The numeric and binary representations for each of the two distinct text fragments of Table 1(c) are depicted in Table 2.

Table 2. Numeric (a) and binary (b) feature representation for the text fragments of Table 1(c), $f_1 = \text{course number}$, $f_2 = \text{course title}$

	f_1	f_2	...
T(25, 25)	2,	0,	...
T(27, 28)	0,	1,	...

(a)

	E_1			E_2		
	f_1	f_2	...	f_1	f_2	...
T(25, 25)	1,	0,	...	1,	0,	...
T(27, 28)	0,	0,	...	0,	1,	...

(b)

4 Experiments

Our goal is to empirically evaluate the proposed architecture in the context of single-slot IE from the Web. For this purpose, we conducted experiments on the task of IE from pages across multiple Web sites, which exhibit multiple formats, including tables, nested tables and lists, thus making the extraction task more difficult.

4.1 Base-learners and meta-learners employed

At base level, we experimented with three learning algorithms for performing IE: STALKER [12], (LP)² [2] and Hidden Markov Models (HMMs) [13]. In this paper we used STALKER in a single-slot mode, as described in [15]. For the HMMs, we adopted the approach proposed in [7] and [14]. For the (LP)² system, we used the default settings of the *Amilcare* [3] environment¹, in which the (LP)² is embedded.

At meta level, we experimented with four classification algorithms, all implemented in the WEKA environment [17]. The first one is *j48*, a reimplementation of the C4.5 decision-tree learning algorithm. The next two belong in the family of boosting algorithms: *AdaBoost.M1* [8], with *j48* as a weak classifier, and *LogitBoost* [9]. The last one is the *IB1*, an implementation of the 1-nearest-neighbor algorithm.

4.2 Dataset description

Experiments were conducted on a collection of 101 Web pages describing CS courses, collected from four different university sites in the context of the *WebKB* project [4]. Three facts were hand-tagged for this domain: *course number*, *course title*, and *course instructor*. All pages were pre-processed by a *tokenizer* module, using *wildcards* [12]².

This corpus was selected due to the fact that it has been used in the past and results are reported in [5]. The approach in [5] is a multi-strategy one, but it does not involve the learning of a meta-classifier. Simple regression models are learned to *map* the relationship between confidence values in the predictions of the base classifiers to true probabilities. At runtime, they rely on a voting scheme, to decide upon the prediction with the highest true probability.

4.3 Results

In order to evaluate our approach we employed a 5-fold *double* cross-validation procedure, known as *cross-validation stacking* [18] and we used *micro-average* recall and precision over all facts. Table 3 shows the base-level experimental results for the CS courses domain. Results for the *F1* metric are also provided, which is the harmonic mean of the recall and precision metrics.

Table 3. Base-level results for the CS courses domain

<i>Macro (%)</i>	<i>Prec.</i>	<i>Recall</i>	<i>F1</i>
HMMs	60,85	62,06	61,45
(LP) ²	69,74	62,12	65,71
STALKER	19,77	49,55	28,26
Best Individ. [5]	74,37	59,47	66,08

¹ The pattern-length was set to 5.

² For the (LP)², pages were preprocessed by a POS tagging module and a Stemming module.

In the CS courses domain, the results of the (LP)² are comparable to the best individual learner’s results, reported in [5]. (LP)² and HMMs share the same recall, however (LP)² achieves a higher precision. STALKER does not perform well in this domain. However, we rely on the diversity in the results of the three systems, aiming at higher performance at meta-level.

Tables 4(a) and 4(b) show the meta-level results, using the *numeric-feature* and *binary-feature* representation respectively. The results are micro-averages of the corresponding results for the three facts. The false class is excluded as it is of no particular interest.

Table 4. Meta-level results using (a) the *numeric-feature* and (b) *binary feature* representation for the CS courses domain

<i>Macro (%)</i>	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>
J48	83,80	59,59	69,65
AdaBoostM1	83,80	59,59	69,65
LogitBoost	84,59	58,93	69,46
KNN (k=1)	84,33	58,75	69,25
Average	84,13	59,22	69,50
M/strategy [5]	N/A	N/A	66,9

(a)

<i>Macro (%)</i>	<i>Prec.</i>	<i>Rec.</i>	<i>F1</i>
J48	86,35	56,45	68,27
AdaBoostM1	86,92	56,15	68,22
LogitBoost	87,48	56,03	68,31
KNN (k=1)	85,19	57,60	68,73
Average	86,49	56,56	68,38
M/strategy [5]	N/A	N/A	66,9

(b)

A clear conclusion from the above results is that the differences between the meta-level classifiers –in each vector representation- are negligible. The *numeric-feature* representation led to slightly better results than the *binary-feature* one, but the difference is too small to lead to an interesting conclusion.

Comparing the meta-level results of Table 4 against the base-level results of Table 3 and the results reported in [5] we note a small decrease in recall, accompanied by a substantial improvement in precision. The meta-level classifiers exploited the diversity in the predictions of the three systems and achieved an overall performance higher than the individual IE systems. The overall conclusion is that the proposed meta-learning framework helps to improve the extraction performance of a series of base-level IE systems.

5 Conclusions

We presented and evaluated a meta-learning framework in the context of IE from the Web. The proposed framework is independent of the employed IE systems at base-level that are not required be classifiers. The presented results are encouraging, showing that the proposed approach improves the precision and overall performance of the IE systems, while outperforming also the state-of-the-art reported in the literature.

Plans for future work include experiments with more complex meta-level vector representations. Additional sources of information (e.g. DOM-based information) will also be investigated. Finally, we plan to experiment with more IE systems and more extraction tasks, in order to evaluate the proposed framework more thoroughly.

References

1. Chan P. K., Stolfo S. J., On the Accuracy of Meta-Learning for Scalable Data Mining. *Journal of Intelligent Information Systems* 8(1): 5-28, (1997).
2. Ciravegna, F., Adaptive Information Extraction from Text by Rule Induction and Generalization. In *Proceedings of the 17th IJCAI Conference*. Seattle (2001).
3. Ciravegna, F., Amilcare: adaptive IE tool, <http://nlp.shef.ac.uk/amilcare/>.
4. Craven, M., DiPasquo, D., Freitag, D., McCallum, A.K., Mitchell, T., Nigam, K., Slattery, S., Learning to extract symbolic knowledge from the World Wide Web, *19th AAAI* (1998).
5. Freitag, D., Machine Learning from Informal Domains, *PhD Thesis*, CMU, (1998).
6. Freitag, D., Kushmerick N., Boosted Wrapper Induction, *17th AAAI Conference*, (2000).
7. Freitag, D., McCallum, A.K., Information Extraction using HMMs and shrinkage, *AAAI-99 Workshop on Machine Learning for Information Extraction*, pp.31-36 (1999).
8. Freund, Y., Shapire, R.E., A Decision-theoretic Generalization of online Learning and an Application to Boosting, *Journal Of Computer and System Sciences*, 55(1), 119-139 (1997)
9. Friedman, J., Hastie, T., Tibshirani, R., Additive Logistic Regression: a Statistical View Of Boosting. *Technical Report*, Stanford University (1999).
10. Hsu, C., Dung, M., Generating Finite-state Transducers for Semi-structured Data Extraction from the Web, *Journal Of Information Systems*, Vol 33 (1998).
11. MUC 7, http://www.itl.nist.gov/iaui/894.02/related_projects/muc.
12. Muslea, I., Minton, S., Knoblock, C., Hierarchical Wrapper Induction for Semistructured Information Sources, *Autonomous Agents and Multi-Agent Systems*, 4:93-114, (2001).
13. Rabiner, L., A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77-2 (1989).
14. Seymore, K., McCallum A.K., Rosenfeld, R., Learning hidden Markov model structure for Information Extraction. *Journal of Intelligent Information Systems* 8(1): 5-28, (1999).
15. Sigletos, G. Paliouras G., Spyropoulos C.D., Hatzopoulos M., Mining Web sites using wrapper induction, named entity recognition and post-processing, *1st European Web Mining Forum*, Cavtat (Dubrovnik) Croatia, September 2003 (to appear).
16. Todorovski, L., Džeroski, S., Combining Classifiers with Meta Decision Trees, *Machine Learning Journal*, Kluwer Academic Publ., Volume 50-(3), p.223-249, (2003).
17. Witten, I., Frank, E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, *Morgan Kaufmann Publishers* (2000).
18. Wolpert, D., Stacked Generalization, *Neural Networks*, 5(2): 241-260 (1992).

ACKNOWLEDGEMENTS

This work has been partially funded by a research grant -provided by the NCSR “Demokritos”- and CROSSMARC, a EC-funded research project.

Information Extraction via Double Classification

An De Sitter
Dept. of Mathematics and Computer Science
University of Antwerp
Middelheimlaan 1
2020 Antwerpen, Belgium
anneleen.desitter@ua.ac.be

Walter Daelemans
CNTS
University of Antwerp
Universiteitsplein 1
2610 Antwerpen, Belgium
daelem@uia.ua.ac.be

Abstract

Information Extraction is concerned with extracting relevant information from a (collection of) documents. We propose an approach consisting of two classification-based machine learning loops. In a first loop we look for the relevant sentences in a document. In the second loop, we perform a word-level classification. We test the system on the Software Jobs corpus and we do an extensive evaluation in which we discuss the influence of the different parameters. Furthermore we show that the type of evaluation method has an important influence on the results.

1 Introduction

An Information Extraction (IE) system has as goal to extract relevant information from a (collection of) document(s). What kind of information is relevant is defined by a template.

In this paper we propose an approach consisting of two classification-based machine learning loops. In the first loop we select the sentences in the document that might contain relevant information. In a second loop we perform a deeper analysis of those relevant sentences by performing word-level classification. By using a rule-based classifier (namely Ripper [3]) for the second loop (unlike a Naive Bayes classifier for the first loop), we obtain human-readable rules.

We did an extensive evaluation of this Double Classification approach. To achieve this, we adapted the evaluation described by Freitag [4]. We test our method on the Software Jobs corpus¹ and study the influence of the parameter settings on the one hand, and the evaluation methodology on the

other hand. We obtain good results on most of the template slots, although for some slots (**language** and **area**) the word-classification approach is not yet capable of finding good rules. Excluding those two slots, we obtain a recall of 77% and a precision of up to 59% if we require that *all occurrences* of an item are extracted. If we use the *one best per document* approach, we obtain recall up to 77% and precision up to 65%. Other IE systems that are evaluated on this corpus (RAPIER [2] and WHISK [12]) obtain a better precision but a worse recall. We also show that the double classification approach indeed improves upon a single word-based classification approach.

The main contributions of this paper are the introduction of the double classification method on the one hand, and on the other hand, an extensive evaluation in which we show that the method of evaluation has an important influence on the results, and therefore on any comparison between systems and approaches. The organization of the paper is as follows: in section 2 we introduce the double classification approach. Section 3 discusses the need for a clear evaluation methodology, section 4 shows the experimental results. Finally, section 5 gives conclusions, related research and further work.

2 Double Classification

The intuition for our approach comes from the observation that humans, when looking for a specific piece of information in a document, don't read the whole document in detail at once. One starts reading globally and based on a quick and superficial analysis then starts reading parts of the text in more depth. It is only in the second phase that the information to be extracted is identified.

¹Downloaded from <ftp://ftp.cs.utexas.edu/pub/mooney/ijoh>

tences. For each sentence we use a Naive Bayes classifier working on a bag-of-words representation of the sentence to decide whether it is relevant or not. By doing this, we get a set of relevant sentences s_1, s_2, \dots per document ordered by a measure of certainty. In a second step, we use a rule-based classifier that decides for each word in the sentence whether it belongs to the entity to be extracted. We can do this for the best, the two best,... or all relevant sentences.

Example 1 *Figure 1 shows an example of a job advertisement from the Software Jobs corpus. The item to be extracted is e.g. the job title. Suppose that a Naive Bayes classifier considers three sentences as relevant, thus having a confidence higher than 0.5. In the second step we may use the best relevant sentence, the two best or all the relevant sentences to do the word-level classification. In section 4 we discuss the consequences of each choice.*

The main motivation for this approach is that the first loop helps to solve the unbalanced training data problem. Suppose we have 100 documents of 40 sentences each. A sentence consists of 10 words. Every document has exactly 1 sentence containing a three-word slot filling. Without the selection of sentences in the first loop, only 0.8% of the words belong to the positive class. Because of this skewed class distribution, a word-based classification approach is infeasible. In our approach however, with the selection of resp. 1 or 3 sentences in each loop, the sizes of the positive classes become respectively 30% and 10%, hence enabling a word-based classification.

We now describe the two classification loops in more detail.

2.1 Sentence classification

In the first loop we want to decide at sentence-level whether a sentence contains relevant information or not. Because we are only doing a deeper analysis of those sentences that this step selects as relevant ones, we need a high recall. In addition, we want the precision to be as high as possible, but this is less crucial than the high recall. For the sentence classification loop, we use a Naive Bayes classifier. We used **Bow** [9], a library of C-code designed and written by Andrew McCallum, and in particular we used **Rainbow**, a front-end that does document classification. **Rainbow** first reads all training data and archives a model containing their statistics. Using this model, **Rainbow** performs classification for each

In this case, each sentence is seen as a separate document. We use the sentences as bags of words without adding any further linguistic information such as POS tags.

The greatest problem we encountered was the large difference between the number of relevant and non-relevant sentences. In our experiments (see section 4) we typically have 1 relevant sentence for each 40 non-relevant ones. This leads to highly unbalanced classes with which standard Naive Bayes has difficulties. As an initial solution, we sample from the set of non-relevant sentences about the same amount of data as we have relevant sentences. For testing purposes we use the data as is (with unbalanced classes). By doing this, we achieve our goal of high recall (we obtain 95-100% recall) and precision of 80-90% at sentence level. So, almost all relevant sentences are recognized, and not too many mistakes are made on the non-relevant sentences.

2.2 Word Classification

The word-level classification is developed in analogy with (NP) chunking. Chunking groups words in sentences into coherent groups such as Nominal Phrases (NPs, e.g. “the old man) and Adverbial Phrases (AdvP, e.g. “not very well). [11] described chunking as a tagging task to be solved with transformation-based error-driven learning. Therefore each word is classified as *in*, *out*, or *at the border* of a particular chunk. In analogy to this, we classify each word of the sentence as being *in* or *out* of the entity to be extracted.

Example 2 *If the entity to be extracted is the job title, sentence 13 from the advertisement in Figure 1 should be annotated as follows ($_I$ stands for in, $_O$ for out the entity): Major_O corporations_O have_O immediate_O openings_O for_O junior_O database_I administrators_I ._O In this representation ‘database administrators’ is unambiguously annotated as the job title to be extracted.*

For the word-level classification, we prefer having human-readable rules, such that we have some insight into how the task has been solved. We used RIPPER [3], a rule-based classification algorithm.

To keep the system easily adaptable to other tasks, we tried to use as little information as possible to solve the task. The default feature vector (per word) consists of the word to be classified, its POS tag, a set of the three words before, a set of the three POS tags before, a set of the three words after, and a set of the three POS tags after the word.

class-before We added the class of the former word. For training we used the actual class of the former word, for testing the predicted class. By doing this, we simulate a simple HMM-approach without losing the advantage of having readable rules.

context Instead of using only the immediate context of a word, we use the whole sentence (words and POS tags), still divided in a set of words (tags) before and a set of words (tags) after the actual word.

attributes We tried adding a few different extra attributes: place in the sentence, and boolean values which indicate if the word starts with a capital, consists of all capitals or contains a digit.

Example 3 For sentence 13 from the job advertisement shown in Figure 1 the class-before feature vector for the word 'developers' is :

WORDS-BEFORE: for Visual Basic

TAGS-BEFORE: IN JJ NNP

WORDS-AFTER: .

TAGS-AFTER: Punc

WORD: developers

TAG: NNS

CLASS-BEFORE: the prediction on the former word.

The word-level classifier was trained only on the relevant sentences.

Figure 2 shows an example of a hypothesis built by RIPPER for extracting **salary** from the Software Jobs corpus. The first rule, for example, can be read as follows: if the class of the word before was 'I', the word itself is '\$' and the tag after the word is 'CD', the current word is tagged as 'I'.

3 Evaluation

For the evaluation of IE systems, typically standard metrics from Information Retrieval are used, namely precision and recall. Precision is the number of correctly predicted entities divided by all entities predicted. Recall is the number of correctly predicted entities divided by the number of entities that should have been found. To compute those and other metrics, we construct a confusion matrix.

In Information Retrieval or in standard classification applications, a confusion matrix is straightforward to build: a document is relevant or not, a mushroom is poisoned or not,... However, in IE the

with manually annotating a text, different annotators will often disagree.

Example 4 Consider the following sentence:

Our company is looking for a senior database administrator (male/female) to lead the team.

there are different possibilities for job title: database administrator, senior database administrator, database administrator (male/female), and senior database administrator (male/female) can all be seen as valid answers by different annotators.

As a result of this, researchers have used various criteria for counting an extracted item as being correct. Some require the exact same boundaries as were manually annotated in the training set, others consider "almost the same" boundaries sufficient.

What criterion has to be used, may depend on the situation as well. If the goal of the IE system is to fill a database on which queries can be asked, one wants to be sure that whatever gets into the database is exact. On the other hand, if someone has as task to point out the job title in each document in a set of job advertisements, he will be helped more by a system that gives a solution overlapping with the correct answer for each document, than by one that returns the exact answer for only a small portion of the documents.

However, to be able to interpret results and to compare different IE systems, it is important to know *how* one has performed the evaluation. To define an evaluation, the first choice that has to be made, is whether we want to find *all occurrences* (AO) of an entity (e.g. every mention of the job title in the advertisements should be found), or whether it suffices to find one occurrence for each template slot. The latter approach is called *one best per document* (OBD).

On the other hand, we have to decide when an extracted entity is counted as a *true positive*. In order to implement the ideas mentioned before, Freitag proposed three basic criteria [4]:

exact The predicted instance matches exactly an actual instance.

contain The predicted instance strictly contains an actual instance, and at most k neighboring tokens.

overlap The predicted instance overlaps an actual instance, there are at most k neighboring to-

We implemented those criteria as three operators, the latter two having 1 (k) and 2 (k, l) parameters.

Example 5 Suppose the sentence in Example 4 is annotated as follows:

Our company is looking for a <title>
senior database administrator
</title> (male/female) to lead the team.

- senior database administrator is correct for the three operators;
- senior database administrator (male/female) is wrong when using the exact operator, is correct using the contain and overlap operators admitting 1 neighboring token.
- database administrator (male/female) is wrong when using the exact or contain operator, and is correct when using the overlap operator and admitting 1 neighboring token and 1 missing token.

In our experiments, we show that the choice between AO and OBD on the one hand, and the type of operator on the other hand, often has a large influence on the results obtained. Thus, it is of utmost importance that the evaluation method is clearly indicated, to be able to interpret the results correctly.

4 Experiments

4.1 Setup

For our experiments we used the Software Jobs corpus: a set of 600 computer-related job postings. Figure 1 shows an example of a job advertisement from the corpus. The templates for this corpus consist of 17 slots: id, country, state, city, company, title, salary, recruiter, post-date, desired years of experience, required years of experience, desired degree, required degree, platform (e.g. Windows NT), application (e.g. SQL server), area and language (e.g. Java)). Several of those slots take multiple fillers (e.g. language, area, ...). Not all entities appear in each document.

For our experiments we used 10-fold cross-validation². We report recall and precision, aver-

²Except for some rare slots as e.g. desired years of experience. In this case we used 5-fold cross-validation to omit

feature vector	AO		OBD	
	recall	precision	recall	precision
class-before	77%	43%	74%	26%
attributes	78%	44%	75%	38%
small context	49%	46%	58%	40%

Table 1: Influence of the feature vector used to extract the “company”. All relevant sentences were used, we report the scores obtained on finding the exact boundaries.

nr sentences	AO		OBD	
	recall	precision	recall	precision
1	18%	33%	33%	37%
2	34%	35%	34%	37%
3	40%	34%	35%	37%
all	43%	31%	35%	35%

Table 2: Influence of the number of relevant sentences used in the word-level classifier to extract the “title”. The feature vector containing the class-before was used, we report the scores obtained on finding the exact boundaries.

aged over those cross-validations. For some overall results, we report F_1 -measure as well. We evaluate the AO as well as the OBD setting. We evaluate each slot separately and report average recall and precision (and F_1 -measure) over all slots as overall scores.

4.2 Results

We report results on two fields: first we evaluate the influence of different parameter settings of the double classification method on recall. Secondly, we report on how the different evaluation criteria influence the results on single slots and on the overall result. Finally, we did some testing with using only the word-level classifier, and show that performance suffers a lot from omitting the sentence-level classifier.

Influence of the parameter settings

The double classification method has two important parameters: the feature vector used for the word-level classification and the number of sentences indicated as relevant by the sentence-level classifier that are used for the word-classifier.

Table 1 shows the influence of the feature vector used to extract **company** from the job advertisement. Early tests(not included in the table) showed

	AO		OBD	
	recall	precision	recall	precision
exact	43%	31%	35%	35%
contains(2)	54%	39%	44%	45%
overlaps(2,1)	84%	61%	70%	71%

Table 6: Influence of the operator used to evaluate the extraction of the “title”. All relevant sentences were used, feature vector is default with class-before.

adds important information. The first row gives the scores when using the class-before feature vector. In the second row, we add extra attributes as described before. In the third row, more context is added to the class-before feature vector. We see that adding extra attributes does improve the precision somewhat in the OBD-case, but doesn’t help otherwise. In this case of extracting ‘company’, adding extra context deteriorates the results as far as recall is concerned, while precision is improving the result somewhat. We tested this on other entities as well and noticed similar results.

Table 2 shows the influence of the number of relevant sentences that are passed from the sentence-level classifier to the word-level classifier, tested in particular for the ‘title’ slot. As might be expected, this has no real influence on recall or precision in the OBD setting, but it has a rather large impact on the recall in the AO setting. By using more sentences, we obtain a higher recall, but when using all relevant sentences, the precision decreases, although not too much.

Table 3 gives an overview of the scores of each template slot, requiring exact boundaries, using the class-before feature vector and all relevant sentences. RIPPER failed to find good rules for the slots **language** and **area** using the current feature vectors, and obtained no better results than the default error (on word-basis). One possible explanation is that those fields mostly take multiple slot fillers which typically are mentioned in the document as an enumeration of some sort. Regular expressions probably would do better for those entities. Further important remarks are:

1. Some entities, e.g. ‘company’ and ‘title’ obtain much better results using more lenient evaluation methods, as is discussed in the next paragraph.
2. ‘state’ as well as ‘city’ obtain a better precision but worse recall by using only the best relevant sentence for the word-classification instead of

Influence of evaluation methodology

In section 3 we discussed that a more lenient evaluation than requiring exact boundaries can be useful. We evaluated all entities with three operators: requiring exact boundaries (*exact*), admitting two neighboring tokens (*contain(2)*), and admitting two neighboring tokens plus missing at most one token (*overlap(2,1)*). Table 6 shows the results of those evaluations for the ‘title’ slot. Recall as well as precision increase a lot by using a more lenient operator. This gives us an important insight in the errors made by the IE system as well: the large gap between *exact* and *overlap*-scores show that the method is capable of finding “almost correct” solutions.

Although not all entities are influenced equally by this, we see important differences in the overall scores. Table 5 gives recall and precision for the overall score. The overall score has been calculated by taking the average over all slots, except for **language** and **area**, because for those results we did not obtain better results than the default error. Two other systems that have been trained on (subsets of) the same data set, RAPIER [2] and WHISK [12], both obtain recall of about 55% and precision of about 85%. They used different methods to count performance, so comparison is difficult, but it still seems clear that although we obtain a significant lower precision, our recall is much higher. A more extensive comparison using exactly the same evaluation method and data should be done to allow more reliable comparisons.

Single level classification

In Table 4 we compare our double classification approach with a baseline system which uses only the word-level classifier. For this baseline system we trained Ripper on approximately 250 (randomly selected) job advertisements, using all (positive as well as negative) sentences. We tested on approximately 60 (randomly selected) job advertisements. Because of memory requirements, it was not possible to use all available data. We did use the class-before setting. We tested the slots **country**, **state**, **salary**, **company** and **title**. The scores for **country** are comparable to the scores obtained by the double classification, **salary** gives worse, but still acceptable results, but the other fields give much worse results than obtained by the double classification approach. For **state** we see that the single classifier is not able to distinguish between

mentions of states. A deeper look at the results for `company` and `title` taught us that although the single classifier is able to find the ‘almost’ place of the entities (`title` obtained 35% precision, 37% recall for *all occurrences*, 30% precision, 27% recall for *one best per document* using the overlap(2,1) measure), it is not capable of finding the correct fillers.

5 Conclusions, related research and further work

We proposed a double classification loop approach to alleviate the problem of unbalanced training data sets in a classification-based learning approach to IE. An important advantage of this method is that by using a rule-based classifier for the second loop, we obtain human-readable extraction rules. We used no semantic information and only POS tags as syntactic information in order to keep the system as portable as possible. Furthermore we did an extensive evaluation in which we discussed the effect of different parameters of the system and showed that it is of utmost importance to be clear about the evaluation methodology used to be able to interpret the results correctly.

Although a hmm approach[1, 5, 6, 10] to IE also uses basically a classification-based approach with a state for words that are part of the filler of a slot to be extracted and a state for the ‘empty tag’, they are limited in the amount of lexical context they can take into account, and in the amount and types of information they can take into account without running into severe sparse data problems. In [13] it is shown that word-classification based learning methods can outperform hmms.

We believe the double classification approach is new in machine learning-based IE. The first, sentence-level, classification loop is similar to machine learning approaches to document summarization, such as [8], in which sentences are classified as relevant or not for appearing in the summary. A two-stage approach in the context of hybrid statistical and knowledge-based IE can be found in [7].

Future research concerns the improvement of the *double classification* approach by extracting different template slots at the same time. Furthermore, we are planning to test this IE system on other domains and compare it to alternative methods. We are also working on a formalization of the evalua-

References

- [1] D. Bikel et al. Nymble: a high-performance learning name-finder. In *Proc. ANLP*, 194–201, 1997.
- [2] M. E. Califf and R. J. Mooney. Bottom-up relational learning of pattern matching rules for information extraction. In <http://citeseer.nj.nec.com/califf02bottomup.html>.
- [3] W. Cohen. Fast effective rule induction. In *Proc. ICML*, 1995.
- [4] D. Freitag. Machine learning for information extraction in informal domains. In *Phd thesis, Carnegie Mellon University, Pittsburgh PA.*, 1998.
- [5] D. Freitag and A. McCallum. Information extraction using hmms and shrinkage. In *Proc. AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999.
- [6] D. Freitag and A. McCallum. Information extraction with hmm structures learned by stochastic optimization. In *Proc. of AAAI*, 2000.
- [7] P. Jacobs, G. Krupka, and L. Rau. Lexico-semantic pattern matching as a companion to parsing in text understanding. In *Fourth DARPA Speech and Natural Language Workshop*, 337–342, 1991.
- [8] J. Kupiec, J. O. Pedersen, and F. Chen. A trainable document summarizer. In *ACM SIGIR*, 68–73, 1995.
- [9] A. McCallum. Bow: a toolkit for statistical language modeling, text retrieval, classification and clustering. In <http://www.cs.cmu.edu/~mccallum/bow>, 1996.
- [10] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proc. ICML*, 2000.
- [11] L. Ramshaw and M. Marcus. Text chunking using transformation-based learning. In *Proc. Workshop on Very Large Corpora, ACL*, 82–94, 1995.
- [12] S. Soderland. Learning information extraction rules for semi-structured and free text. In *Machine Learning 34*, 233–272, 1999.
- [13] J. Zavrel and W. Daelemans. Feature-rich memory-based classification for shallow nlp and information extraction. In *Text Mining. Theoretical aspects and applications. Springer LCNS series.*, 2003.

```

1.  From : spectrum@onramp.net
2.  Newsgroups : austin.jobs
3.  Subject : <country> US </country> - <state> TX </state> - <city> Austin
    </city>
    - <language> VISUAL BASIC </language> <title> Developers </title>
    <salary> 50Kto 70K </salary> Date : Sat , <post-date> 23 Aug 97
    </post-date> 09 : 52 : 21
4.  Organization : OnRamp Technologies , Inc. ; ISP
5.  Lines : 65
6.  Message - ID : < <id> NEWTNews.872347949.11738.consultswn - n </id> >
    NNTP - Posting - Host : ppp10 - 28.dllstx.onramp.net
    ...      ...      ...
10. <country> US </country> - <state> TX </state> - <city> Austin </city>
    - junior <title> database administrators </title>
    <salary> 50Kto 70K </salary>
11. POSTING I.D .
12. D05
13. Major corporations have immediate openings for junior
    <title> database administrators </title> .
14. <req-years-experience> 2 </req-years-experience>
    - <desired-years-experience> 5 </desired-years-experience> years
    experience ; <application> Oracle </application> or <application> SQL
    Server </application> helpful .
15. <platform> Windows 95 </platform> and <platform> Windows NT
    </platform> programming a plus .
16. Please contact Bill Owens at ( 972 ) 484 - 9330 ; FAX ( 972 ) 243 - 0120 at
    <recruiter> Resource Spectrum </recruiter> .
    ...      ...      ...
26. <recruiter> Resource Spectrum </recruiter>
27. 5050 Quorum Dr. , Ste 700
28. Dallas , Texas 75240
    ...      ...      ...

```

Figure 1: (Part of) an annotated example of a job advertisement.

```

i :- CLASS_BEFORE=i, TAGS_AFTER  CD, WORD  '$'.
i :- CLASS_BEFORE=i, TAG  CD.
i :- CLASS_BEFORE=i, TAGS_AFTER  CD, WORDS_BEFORE  ':''.
i :- WORD  '$'.
i :- CLASS_BEFORE=i, TAGS_AFTER  CD, WORDS_AFTER  '000'.
i :- CLASS_BEFORE=i, TAG  NN, WORD  '-'.
i :- TAG  TO.
i :- CLASS_BEFORE=i, WORDS_AFTER  ':'', WORDS_BEFORE  '$'.
i :- CLASS_BEFORE=i, WORD  '-'.
i :- CLASS_BEFORE=i, WORD  hr.
i :- CLASS_BEFORE=i, WORDS_BEFORE  '- ', WORDS_AFTER  ';''.
i :- CLASS_BEFORE=i, WORD  '55K'.
i :- CLASS_BEFORE=i, TAG  NN, TAGS_AFTER  IN.
i :- WORDS_BEFORE  Salary, WORDS_BEFORE  ':'', WORDS_AFTER  '- '.
i :- WORDS_BEFORE  in, WORDS_AFTER  '.''.
i :- CLASS_BEFORE=i, WORDS_AFTER  and.
i :- WORDS_AFTER  '70K', TAG  CD.
default o.

```

	AO			OBD		
	recall	precision	F_1	recall	precision	F_1
ID	98%	97%	97%	96%	99%	97%
country	98%	92%	95%	94%	91%	92%
state	97%	77%	86%	95%	93%	94%
city	95%	84%	89%	92%	90%	91%
company	78%	45%	57%	74%	26%	38%
title	43%	31%	36%	35%	35%	35%
salary	70%	56%	62%	72%	62%	67%
recruiter	79%	40%	53%	74%	44%	55%
post-date	99%	84%	91%	97%	99%	98%
desired degree	45%	28%	35%	37%	29%	33%
required degree	43%	29%	35%	51%	41%	45%
desired years experience	55%	33%	41%	66%	36%	47%
required years experience	80%	50%	62%	81%	72%	76%
platform	34%	31%	32%	38%	35%	36%
application	29%	32%	30%	30%	31%	30%
area	17%	16%	16%	18%	16%	17%
language	27%	25%	26%	34%	33%	33%
overall	63.9%	49.9%	55.5%	63.7%	54.8%	58.0%

Table 3: Individual results for each entity. All relevant sentences were used, feature vector is default with class-before, exact boundaries were required.

	AO			OBD		
	single	double best relevant	double all relevant	single	double best positive	double all positive
company	6%/4%	41%/22%	49%/77%	0%/0%	40%/80%	26%/74%
country	87%/92%	77%/97%	96%/49%	90%/87%	91%/94%	97%/93%
state	23%/23%	93%/95%	97%/93%	22%/22%	93%/95%	97%/93%
title	9%/10%	33%/18%	31%/43%	3%/3%	38%/33%	35%/35%
salary	55%/55%	56%/70%	64%/54%	46%/48%	62%/72%	68%/72%

Table 4: Comparison between single loop information extraction (only word-based classification using Ripper) and the double classification approach. For the latter approach we report results from using the best relevant sentence and using all relevant sentences for the word-level classification. We report (precision/recall).

	AO			OBD		
	recall	precision	F_1	recall	precision	F_1
exact	70%	54%	61%	70%	59%	64%
contains(2)	73%	55%	63%	72%	60%	65%
overlaps(2,1)	77%	59%	67%	77%	65%	70%

Table 5: Influence of the operator used to evaluate on the overall score, minus **language** and **area**. All relevant sentences were used, feature vector is default with class-before.

Information Extraction as a Semantic Web Technology: Requirements and Promises

Mark Stevenson and Fabio Ciravegna

Department of Computer Science,
211 Regent Court, Portobello Street,
Sheffield S1 4DP
United Kingdom
{marks,fabio}@dcs.shef.ac.uk

Abstract

The Semantic Web will require services to annotate web pages with the necessary meta-data. Information Extraction (IE) may be a suitable technology for this purpose. This paper discusses the requirements generated for applications of IE to the Semantic Web and the degree to which current technology meets them.

1 Introduction

The Semantic Web (Berners-Lee et al., 2001) is an ongoing initiative to make the World Wide Web a more useful resource by standardizing the descriptions of available information and services. It is expected that this will be achieved by adding various forms of meta-data to web pages and, while this may be feasible for newly created pages, it is unlikely to be carried out for existing web content. The only reasonable approach is to annotate these pages automatically. Information Extraction (IE) may be a suitable technology for automating this process (Ciravegna, 2003).

Research in IE has been largely driven by the Message Understanding Conferences (MUC) (MUC, 1991 1993 1995 1997 1998). These exercises focused on identifying information from free text such as newswire stories. The participants were required to identify every item of a specific semantic type in the text, a process known as Named Entity (NE) recognition. For example in the sixth MUC the semantic types included **PERSON**, **ORGANIZATION** and **LOCATION**. Participants were also required to identify specific relations between these entities and combine them into templates where appropriate. The majority of IE systems carried out the stages of NE recognition and relation extraction as separate processes.

It is possible to define some basic requirements for an IE system to be useful for the Semantic Web; these pertain to issues portability, both to domain and extraction task. It is expected that the Semantic Web will be based on many small ontological components (Hendler, 2001) rather than large, complex ontologies like CYC (Lenat, 1995). These components will be continuously extended, merged or created, therefore the annotation services associated with them will have to be constantly adjusted or revised according to these changes. This poses a number of obvious constraints and requirements on the technology to support annotation in terms of usability, portability and maintainability that we will list in the next sections. If IE technology is used to support annotation then new applications will be required whenever a new ontological component is created. Machine learning offer techniques to adapt IE systems to new domains and extraction tasks and for this reason this paper focuses on these approaches.

Two types of resources must be dealt with in order to port an IE system: linguistic resources, such as tokenizers, part of speech taggers and parsers, and semantic resources like gazetteers and ontologies. Creators of Semantic Web ontologies cannot be expected to have expert knowledge of IE and so any annotation tools must be portable by a person with limited IE skills. This problem has already been discussed by Ciravegna (2001). Consequently, methodologies for learning linguistic rules will be needed. In principle the necessary semantic resources could be provided by the ontology. Unfortunately, the kind of ontology defined will satisfy the needs of the customer service, not of the IE component. This means that it will not be linguistically oriented and there is the

possibility that it will not be useful for IE purposes. For example some distinction between two concepts could require deep reasoning on background knowledge which may be beyond the IE capabilities. Moreover some relations could be included that from a linguistic point of view require intermediate representation and reasoning (e.g. metonymic reasoning). This kind of information, if not provided to the IE system, could make the IE task quite complex if not infeasible. Some intermediate level of linguistically oriented ontology definition will be needed. Methodologies derived from the field of ontology learning (e.g. (Maedche and Staab, 2000), (Brewster et al., 2001)) could help in suggesting appropriate representations to non-linguistically-aware users.

The remainder of this paper discusses the suitability of current IE technology for requirements generated by the Semantic Web. Named entity identification and relation extraction technologies are discussed in Sections 2 and 3 respectively. Conclusions are presented in Section 4.

2 Named Entity Identification

Machine Learning techniques have proved to be very popular for named entity identification. Unfortunately, many systems require large amounts of training data to be ported to a new NE tasks. For example, BBN's SIFT system requires the annotation of a training corpus of 790,000 words in order to obtain 90% F-measure on the MUC7 task (Miller et al., 1998). Approaches such as Borthwick et al. (1998) and Mikheev et al. (1999) reduce the burden on the application developer by generalizing from the annotated text, seed rules or example names provided by the user. Riloff (1993) developed a system, AutoSlog, which learned from annotated text to generate semantic lexicons which could be used to identify NEs. Riloff and Shoen (1995) eliminated the need for annotations or rules in an extension of AutoSlog which only required the user to classify texts as relevant or irrelevant for the extraction task. Collins and Singer (1999) reduced the effort required further by using a bootstrapping algorithm which learned from just seven seed rules.

However, each of these approaches is very limited in terms of number of NE types and have often been restricted to those used in the

MUC evaluations. These are generic and domain independent tags since the ontology used was both restricted and flat. It is expected that ontologies used in the Semantic Web will be significantly more complex, containing dozens of domain-specific concepts, instead of the seven used in the later MUC evaluations. These domain specific concepts will occur with less frequency than MUC-style ones which leads to the problem of data sparseness, making supervised learning approaches less feasible. Consequently this technology may not meet the Semantic Web requirements.

Some other approaches, specifically designed for use on the Web, use the regularity of the Web to learn entities contained in web pages. Brin (1998) uses a handful of user-defined examples to bootstrap learning for a task on finding book titles and authors. Ciravegna et al. (2003) employ multiple strategies to bootstrap learning on consistent repositories (e.g. Web sites). These approaches are promising since they remove much of the burden of manual annotation while still delivering good annotation services.

3 Relation Extraction

Simple recognition of entities is unlikely to generate complex enough meta-data for the Semantic Web and so identification of relations is considered to be important (Handschuh et al., 2002). From the IE point of view, relation extraction is a complex task which has not been studied in as much depth as NE recognition. The majority of MUC systems approached the relation extraction task using knowledge engineering approaches which relied on (para)linguistic rules manually created by an expert. The effort required to port these systems to a new domain or extraction task was often considerable, for example, the University of Massachusetts entered a system for the third MUC which required around 1,500 person-hours of expert labour to adapt the system for that extraction task (Lehnert et al., 1992). This overhead makes the knowledge engineering approach infeasible for the Semantic Web. A few systems have tackled this limitation using ML techniques.

WHISK (Soderland, 1999) is a system which learns extraction patterns directly from shallow parsed or unannotated text. WHISK assumes

that the text has already been marked with named entities. Extraction patterns match directly to identify those which are part of a particular relation. The patterns can be applied to text which is either unannotated or partially parsed. This flexibility allowed the system to be applied to a wide range of text types including formal text and web pages. Evaluation was carried out using a simplified version of the management succession task used in the sixth MUC. WHISK was required only to identify relations which were described within a single sentence rather than across texts. The algorithm achieved an F-measure of 55.5. Chieu and Ng (2002) recast the relation extraction problem as a classification task. A score is computed for each pair of entities which occur in the same sentence to determine whether or not they represent a true relation. A maximum entropy learning algorithm was used to determine the score and pairs combined to form a template. Chieu and Ng reported an F-measure of 59.2 using Soderland's evaluation scheme. Yangarber et al. (2000) presented an unsupervised approach to relation learning. Text was pre-processed by examining the output from a parser to identify subject-verb-object tuples, for example **person-resigned-company** and **company-fired-person**. The user provides a set of seed patterns which are then generalized by substituting some elements with wildcards. Each of the patterns which occur in the corpus and match one of the generalized patterns are then evaluated and one chosen to be added to the pattern set. This approach was evaluated in terms of document relevance which makes comparison with other approaches difficult.

A crucial elements in applying ML to relation extraction is to find a way of generalizing patterns in a satisfactory way. Soderland (1999) and Yangarber et al. (2000) generalize patterns by removing restrictions on some elements of their patterns and then searching the corpus for instances which match the relaxed pattern. However, WordNet (Fellbaum, 1998) has been used to generalize patterns in a linguistically principled way (for example (Català et al., 2000)). One of the useful sources of information in WordNet is the lists of synonyms for terms and these could be used to generalize patterns. A potential problem is the fact

that WordNet contains several senses for many of the words which might be of interest to an IE system. For example, there are nine senses for the verb "fire" and only one contains synonyms useful for generalizing patterns for the management succession task (e.g. "dismiss", "sack", "terminate"). Català et al. (2000) avoided this problem by requiring the user to identify the correct entry in WordNet when defining the extraction templates to be filled but this is a burden on the user. Chai and Biermann (1999) used word sense disambiguation to identify the correct WordNet entry. Performance of their IE system improved from an F-measure of 61.7 to 69.2 when disambiguation was used to guide the generalization process.

3.1 Suitability of technology

All the systems mentioned above require preliminary parsing. A parser processes the input so to produce formats more suitable for learning than unrestricted text and may resolve some of the ambiguity in language prior to learning. For example, the Connexor parser used by (Yangarber et al., 2000) analyses active and passive sentences to identify the semantic subject and object. So, for example, "The board fired Jones." and "Jones was fired by the board." produce the same triple (**board-fire-jones**).

Documents to be annotated for the Semantic Web will be of different types, from free texts (largely parsable) to very structured ones, where the information is largely carried by extralinguistic clues (e.g. HTML tags) and therefore largely unparsable. We have also noted how very often mixed types of documents can be found where some parts are highly structured and others consist of free text (Ciravegna, 2003). In this case generic linguistic methodologies (e.g. parsing) will not work properly and therefore the system will not be able to extract information. A system which can operate with or without the need for an intermediate representation is Soderland's WHISK (Soderland, 1999) which can learn patterns which match directly to unparsed text. However, it was found that this approach is more suitable for semi-structured text which is generally syntactically simpler and more regular than free text.

Moreover most of the technology mentioned above is not able to exploit the available ontological information, for example for general-

izing rules and apply reasoning. With the potentially reach ontologies available for the Semantic Web, this could be a relevant limitation. New methodologies are needed for IE able to exploit such information, with the caveat mentioned above, though, that such ontologies could be far from linguistically oriented.

All in all, the systems mentioned above are (very similar to) standard MUC systems where just the step of extracting relations from sentences are performed automatically. These systems still require IE experts for porting the other modules, so - we believe - are not suitable for use in the Semantic Web.

4 Conclusions

This paper has discussed some requirements for the suitability of IE as annotation support for the Semantic Web. The core requirements are portability by non-experts, ability to cope with different text types, ability to use ontological information and ability to train with limited annotated documents. We have discussed the extent to which current IE technology meets these requirements. It was found that current technology is limited in the following respects:

1. the ability to train with a limited amount of material.
2. the ability to learn relations without relying on deep linguistic annotation; the system should be able to exploit linguistic information when existent and reliable but rely on shallower methods when necessary.
3. the ability to use ontological information when available.

Some of these limitations have been addressed by various approaches to IE. For example, Yangarber et al. (2000) and Collins and Singer (1999) use unsupervised learning algorithms to reduce the input required from the user to a few seed patterns and this is an attempt to overcome limitation 1. Limitation 2 is addressed by the WHISK system (Soderland, 1999) which has the ability to learn patterns from both parsed and unparsed text. The final limitation has not really been addressed; while some approaches have made use of linguistic ontologies (see Section 3) these are different from the ontologies

which are expected to be used for the Semantic Web.

In conclusion we believe that IE is a very promising technology for annotation for the Semantic Web. Potentially IE systems could become in the future Semantic Web as important as indexing systems are for search engines in the current for of the Web. In order to get this opportunity, some very focused research effort is needed that goes beyond the usual definitions and limitations of IE as derived from the MUC conferences.

Acknowledgments

This work was carried out within the European Commission framework V project dot.kom (IST-2001-34038). Further information is available from <http://www.dot-kom.org>

The authors are grateful for a number of interesting discussions on the topics in this paper with the dot.kom partners.

References

- T. Berners-Lee, J. Hendler, and O. Lassila. 2001. The Semantic Web. *Scientific American*, 28(5):34–43.
- A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. 1998. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 152–160, Montreal, Canada.
- C. Brewster, F. Ciravegna, and Y. Wilks. 2001. User-centred ontology learning for knowledge management. In *Proceedings of the 7th International Conference on Applications of Natural Language to Information Systems*, pages 203–207, Stockholm, Sweden.
- S. Brin. 1998. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, pages 172–183, Valencia, Spain.
- N. Català, N. Castell, and M. Martin. 2000. ESSENCE: A Portable Methodology for Acquiring Information Extraction Patterns. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 411–415, Berlin, Germany.
- J. Chai and A. Biermann. 1999. The use of word sense disambiguation in an informa-

- tion extraction system. In *Proceedings of the Eleventh Annual Conference on Innovative Applications of Artificial Intelligence*, pages 850–855, Portland, OR.
- H. Chieu and H. Ng. 2002. A maximum entropy approach to information extraction from semi-structured and free text. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence (AAAI-02)*, pages 768–791, Edmonton, Canada.
- F. Ciravegna, A. Dingli, D. Guthrie, and Y. Wilks. 2003. Integrating information to bootstrap information extraction from web sites. In *Proceedings of the IJCAI 2003 Workshop on Information Integration on the Web*. Acapulco, Mexico.
- F. Ciravegna. 2001. Challenges in information extraction from text for knowledge management. *IEEE Intelligent Systems and Their Applications*, 27:97–111.
- F. Ciravegna. 2003. Designing adaptive information extraction for the Semantic Web in Amilcare. In S. Handschuh and S. Staab, editors, *Annotation for the Semantic Web*.
- M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110, College Park, MA.
- C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database and some of its Applications*. MIT Press, Cambridge, MA.
- S. Handschuh, S. Staab, and F. Ciravegna. 2002. S-CREAM - Semi-automatic CREation of Metadata. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW-02)*, Sigüenza, Spain.
- J. Hendler. 2001. Agents and the semantic web. *IEEE Intelligent Systems Journal*, 16(2):30–37.
- W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. 1992. University of massachusetts: Description of the CIRCUS system used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 282–288, San Francisco, CA.
- D. Lenat. 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.
- A. Maedche and S. Staab. 2000. Discovering conceptual relations from text. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 321–325. Berlin, Germany.
- A. Mikheev, M. Moens, and C. Grover. 1999. Named entity recognition without gazeteers. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, Bergen, Norway.
- S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, and R. Weischedel. 1998. BBN: Description of the SIFT system as used for MUC7. In *Proceedings of the 7th Message Understanding Conference*, Fairfax, VA.
- 1991, 1993, 1995, 1997, 1998. *Proceedings of the Third, Fourth, Fifth, Sixth and Seventh Message Understanding Conferences*. Morgan Kaufmann.
- E. Riloff and J. Shoen. 1995. Automatically acquiring conceptual patterns without an annotated corpus. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 148–161, Somerset, NJ.
- E. Riloff. 1993. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 811–816, Washington, DC.
- S. Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 31(1-3):233–272.
- R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. 2000. Unsupervised discovery of scenario-level patterns for information extraction. In *Proceedings of the Applied Natural Language Processing Conference (ANLP 2000)*, pages 282–289, Seattle, WA.

Finding Educational Resources on the Web: Exploiting Automatic Extraction of Metadata

Cynthia Thompson
School of Computing,
University of Utah
Salt Lake City, UT 84112

**Joseph Smarr and Huy Nguyen and
Christopher D. Manning**
Dept of Computer Science, Stanford University,
Stanford CA 94305-9040

Abstract

Searching for educational resources on the web would be greatly facilitated by the availability of classificatory metadata, but most web educational resources put up by faculty members provide no such metadata. We explore using text classification and information extraction techniques to automatically gather such metadata. Text classifications orthogonal to topic matter appear possible with high ($> 90\%$) accuracy, and exact-match information extraction has an F measure around 50%.

1 Introduction

In order to be able to do semantically rich queries over distributed heterogeneous data collections like the web, a key tool is the use of metadata to explicitly annotate documents with relevant information. This is the general goal of the semantic web [1], and such markup schemes exist for education resources, for example, IEEE LOM [2]. In the particular context of the Edutella project [3], learning resources such as lesson plans, tutorials, assignments, and so on are annotated with the educational topic to which they pertain, the education level of the intended audience, and so on. The presence of such attributes allows highly customized searches, as well as quick summaries of available documents.

A major challenge to building a metadata-rich repository is that someone has to manually annotate all the documents. This is a slow and costly process, and many producers of educational content are probably not interested in going back and annotating all their work. In the Edutella context, semantic metadata is available for documents within the Edutella peer-to-peer network, but it would be useful to be able to conveniently access the mass of other educational resources available on the web: there are numerous valuable educational resources available, but finding them using traditional keyword searches is hard, and most of them are not annotated with any useful metadata.

While searches can use available metadata when present, there is thus a clear need to develop tools that can perform some or all of this annotation automatically. Such tools would save content creators time and

would allow content consumers to utilize the web as if it were part of the same metadata rich environment to which they were accustomed. In cases where fully automatic annotation cannot be accomplished with sufficient accuracy, there is still value in providing suggestions to human annotators. Or, one could highlight information that is relevant to the annotation decision, such as word features that a classifier has found to be relevant in text classification discrimination.

There are two major technical avenues for automatic metadata extraction. First is the classification of documents into appropriate categories on various dimensions (e.g., what language the document is written in, what type of learning resource it is, or what level of student it is intended for). Second is the extraction of text from documents for summary fields (e.g., title and author of the document, topics covered in a course description, or readings assigned on a syllabus). In both cases, it is reasonable to seek systems that work from limited training data: for any fine-grained topic there is only a limited amount of material on the web, and if people have to find and annotate most of those pages, then there is little to be gained. So methods that can quickly generalize are of particular interest. When dealing with web pages, another interesting question is whether HTML markup can be usefully exploited in addition to the text content for classification or extraction.

In this paper, we present some early results on this task of providing metadata for educational web pages, considering first text classification, and then information extraction.

2 Data Collection and Annotation

We downloaded text web pages for both classification and information extraction. While it would be useful to extend analysis to other formats such as PDF and postscript, the present experiments used just HTML pages. For classification, we collected 4 different types of resources: syllabi, assignments, tutorials, and exams. Such a text classification task is orthogonal to typical topic-based classification decisions, but it seems reasonable that good results should still be possible based on features present in the documents.

When collecting pages, we restricted ourselves to artificial intelligence and machine learning courses, since the syllabi in this case were also used for infor-

Table 1: Reachable Relevant Pages out of Top 20

Search Term	Num Returned	Num Relevant
syllabus	20.3K	15
class	442K	10
course	550K	7
introduction	552K	7

mation extraction (see below). For the most part, we found these pages by a web search engine using the terms “artificial intelligence” or “machine learning” with “course,” “class,” or “syllabus.” We also used some directories and lists of such courses of which we were aware. We then proceeded from the main course page to find connected pages with assignments or on-line exams. For the tutorials, we used similar search terms and directories.

As noted previously, finding these resources via keyword search alone is difficult. We measured informally the difficulty of the task. We combined the search term “artificial intelligence” with each of “course,” “class,” “syllabus,” and “introduction,” and measured the number of useful pages that were easily reachable from the pages of the top 20 search terms returned. Table 1 summarizes the results. Even these figures are fairly lenient, as we included in the count both duplicates between the four terms and pages that we could find by following one or two links from the page returned by the search. An automated crawler would fare much worse in distinguishing the desirable pages.

Finally, for the extraction task, we took the syllabi collected above and used an annotation tool to tag them with a set of 5 tags, course number, course title, instructor, year, and readings. This tool also removed certain HTML material, such as SCRIPT blocks and comments; these features were also removed from the pages before classification was applied.

3 Classification

For the classification task, we used a classifier to distinguish between different resource types. Classification was done with a maximum entropy classifier, which here used just word-category features, and hence is simply a multiclass logistic regression model. The model uses a Gaussian prior on feature weights for regularization, and is fit by conjugate gradient search. The model is essentially similar to [4], and actually uses the classifier within the sequence model described in [5].

We collected 385 pages in all: 131 assignment pages, 219 syllabus pages, 22 tutorial pages, and 13 exam pages. Given this data distribution, we are close to dealing with a two-class classification problem, so we looked at accuracy both for the two class case and for the more unbalanced data set. So for both cases, for five random splits of the data, we trained the classifier on 80% of the pages and tested on the remaining 20%. The training set accuracy on these splits was

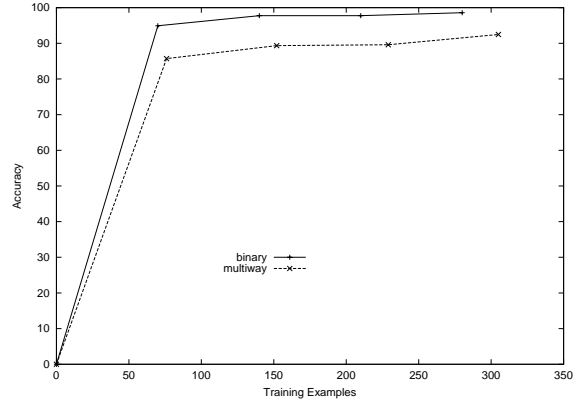


Figure 1: Classification Accuracy for Resource Type

Table 2: Multi-way accuracy per label

Category	Prec	Rec	F ₁
assignment	89.6	91.6	91
syllabus	95.3	100	98
exam	50	27.5	35
tutorial	68.75	51.78	59

uniformly 100% accurate. We created learning curves for both cases, training on increasingly larger portions of the training set, and testing classification accuracy on the test set. The trends are shown in Figure 1. For the full training set in the multi-class case, the average test set accuracy was 92.5% and the number of word features was about 22,160 on average. For the binary case, the average accuracy was 98.6% and the number of word features was 20,235.

For the binary case, errors are fairly evenly split between mistaking **assignment** for **syllabus** and vice versa, with a slight tendency in some splits to mislabel **assignment** as **syllabus** more than the reverse mistake, but both types of mistakes were made in at least one training set size for every split. For the multi-class case, the results were more mixed. The precision, recall, and F₁ are shown in Table 2. For **exam**, all errors were due to their incorrect labeling as **assignment**. This is not surprising given the similar nature of the two. For **tutorial**, they either got mislabeled as **assignment** or **syllabus**, almost equally, but never as **exam**. We examined the weights learned in two of the splits: for **assignment**, the maximum weighted feature in both splits was *forbidden*, coming from statements prohibiting copying from other students or on-line sources!

4 Information Extraction

For many types of metadata, including course titles and instructors, the possible values for fields are not confined to a closed set, and are therefore beyond the extraction capabilities of classification. Information extraction is a promising alternative, since it allows us to accommodate variation in the values of the field,

as well as exploit the surrounding context for extraction. Our general approach to information extraction is the use of class HMMs, in the general tradition of [6]. In this model, the mathematics of which is clearly described in [7], each hidden state generates not only a word, but also a class label: the name of a field to be extracted or ‘Background’. At training time, the state sequence is partially constrained by observed class labels, but not fully determined, and parameter estimation is done by the EM algorithm. For unknown words, the model uses a class-based model, based on features of words, such as capitalization and the presence of numbers. Figure 2 indicates one kind of HMM topology used in the experiments. This structure seeks to extract a single target field, uses a unigram model for the background, and attempts to model the target prefix and suffix with three states. Not shown are self-loops on every state, and forward arcs on the target states. We experimented some with target chains of different lengths based on the field being extracted, but since skipping is allowed in target chains, a longer chain structure can still model targets of shorter lengths, and this parameter did not have much effect.

We also experimented with a single large HMM which contained states corresponding to all of the target fields. The beginning and end of these target chains were fully connected to each other and eight background states (an ergodic context model), with parameter estimation used to find a suitable model structure.

Table 3 summarizes our experimental results. As discussed above, the data set was 219 syllabi coded for 5 fields, and we present the average of 10-fold cross-validation experiments (each test fold is quite small, and there is quite high variance in the results between folds). We used the same evaluation metric as [6] – reporting the F_1 score (harmonic mean) of precision and recall, based on exact target matching, calculated on the basis of correctly instantiating the fields of a metadata relation for the page. This is a fairly stringent evaluation criterion (for instance, a mostly correct evaluation missing a word gets no credit).

Results are in general promising, but some fields are easier than others. Incorporating basic HTML markup tags boosted performance. This is not surprising, as certain HTML tags, such as <title> are strongly correlated with target fields, and targets frequently occur within <i> and tags, or following tags. The two chain lengths are for the length of the target chain, and of each of the prefix and suffix chains. Initially target lengths were chosen heuristically based on the complexity of the fields, but in retrospect simply choosing a uniform chain length of 4 would have made no real difference (given that skipping forward is allowed in target chains). The context chain length partly determines how much context can be modeled (but note that the context states have self-loops, unlike those of [6] – something that we have found useful in other experiments). Here, the data set is sparse enough that having more than one context state on each side of the target does not seem to have any useful discrimi-

Table 3: HMM information extraction results.

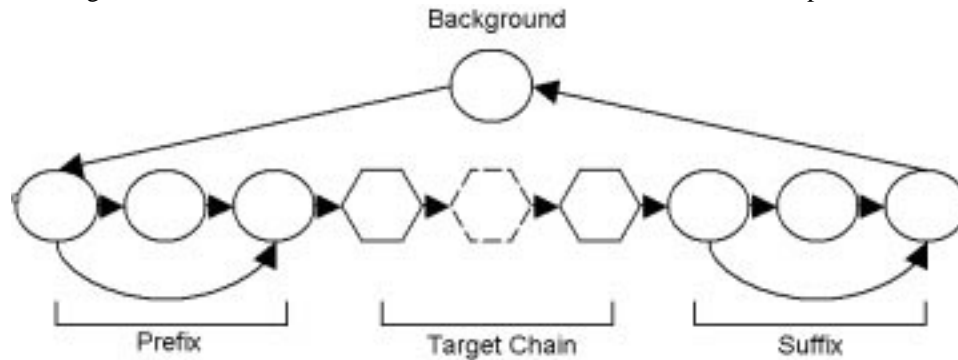
Target Field	Targ/Conx Chain Len	Keep HTML?	First/Best	F_1
<i>One field at a time results</i>				
Course number	4/1	Yes	First	78.3
Course number	2/3	Yes	First	68.8
Course number	4/1	Yes	Best	67.0
Course number	2/3	Yes	Best	63.5
Course number	2/3	No	Best	51.3
Course title	4/3	Yes	First	43.6
Course title	4/3	Yes	Best	52.5
Course title	4/3	No	Best	37.3
Instructor	3/3	Yes	Best	37.0
Instructor	4/1	Yes	Best	35.1
Instructor	3/3	No	Best	35.5
Date	4/1	Yes	Best	53.0
Date	4/3	Yes	Best	51.2
Reading	4/3	Yes	Best	21.0
<i>All fields at once results</i>				
Course number	4/NA	Yes	Best	55.5
Course title	4/NA	Yes	Best	53.2
Instructor	4/NA	Yes	Best	37.8
Date	4/NA	Yes	Best	49.7
Readings	4/NA	Yes	Best	31.1

nating power. We also recognized that certain fields, such as course number and course title, tend to appear near the top of the page. We attempted to exploit this (non-stationary) domain knowledge by additionally trying extracting the first segment labeled as a target, whereas the standard system returns the “best” segment (the one with the highest length-normalized generation probability within a window). This met with mixed success: it was very helpful for the course-number field, but didn’t have positive value for the course-title field. Ways of choosing between targets picked out by the HMM deserves further thought. Finally, the all-fields-at-once HMM might be hoped to do better global modeling of the sequence of entities in a document, at the cost of having a less detailed model of the prefix and suffix contexts of individual fields. But at least for this data set, the two models seem to give roughly equal results overall.

5 Plans for Future Work

The results indicate that automatic extraction of metadata is feasible at least in certain cases, but much could be done to improve the utility of such an approach. Extracting summary information is clearly a more difficult task in general than classifying a page into one of a set of categories. Additional data would presumably help, though in many cases it would be unreasonable to expect more labeled data than was available here. One promising avenue is to exploit existing domain knowledge top-down to constrain classification and extrac-

Figure 2: Indicative HMM structure used in information extraction experiments.



tion. For example, if one knows the course numbering system at a university, that can be helpful in determining whether a course web page is intended primarily for undergraduates or graduate students. Another is to realize that word level data is likely to be too sparse for effective training in many cases, and to make more use of higher level notions, such as person names, which could be provided by a generic named entity recognizer. Eventually, the success metric of this approach is to be measured by whether it saves human annotators time reaching a given standard of annotation quality for documents, and whether it provides value in obtaining educational material from the web beyond simply using a search engine.

References

- [1] Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* **284** (2001) 35–43
- [2] IEEE: Draft standard for learning technology – Learning Object Metadata – ISO/IEC 11404. Technical Report IEEE P1484.12.2/D1 (2002)
- [3] Simon, B., Miklós, Z., Nejdí, W., Sintek, M., Salvachua, J.: Elena: A mediation infrastructure for educational services. In: *Twelfth International World Wide Web Conference*. (2003)
- [4] Zhang, T., Oles, F.J.: Text categorization based on regularized linear classification methods. *Information Retrieval* **4** (2001) 5–31
- [5] Klein, D., Smarr, J., Nguyen, H., Manning, C.D.: Named entity recognition with character-level models. In: *CoNLL 7*. (2003) 180–183
- [6] Freitag, D., McCallum, A.: Information extraction with HMM structures learned by stochastic optimization. In: *Proceedings of AAAI*. (2000) 584–589
- [7] Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, Cambridge (1998)

Semantic Relations in Concept-Based Cross-Language Medical Information Retrieval

Špela Vintar, Paul Buitelaar
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken, Germany
{vintar, paulb}@dfki.de

Martin Volk
University of Stockholm
Department of Linguistics
Universitetsvägen 10C
S-10691 Stockholm, Sweden
volk@ling.su.se

Abstract

We explore and evaluate the usefulness of semantic annotation, particularly semantic relations, in cross-language information retrieval in the medical domain. As the baseline for automatic semantic annotation we use UMLS, which specifies semantic relations between medical concepts. We developed two methods to improve the accuracy and yield of relations in CLIR: a method for relation filtering and a method to discover new relation instances. Both techniques were applied to a corpus of English and German medical abstracts and evaluated for their efficiency in CLIR. Results show that filtering reduces recall without significant increase in precision, while discovery of new relation instances indeed proved a successful method to improve retrieval.

1 Introduction

The aim of Cross-Language Information Retrieval (CLIR) is to find documents in a large, possibly multilingual, collection that are most relevant for a given query, where the language of the query may be different from the language of the documents retrieved. Methods typically used to overcome this language barrier may be divided into: approaches based on bilingual dictionary look-up or Machine Translation (MT) (Hull and Grefenstette, 1996;

Kraaij and Hiemstra, 1998; Oard, 1998); corpus-based approaches utilizing a range of IR-specific statistical measures (Carbonell et al., 1997; Qui, 1995); and concept-driven approaches, which exploit semantic resources (thesauri) to bridge the gap between surface linguistic form and meaning (see Section 6).

The appeal of concept-based approaches is that, in contrast with translation or corpus-based methods, they use linguistic processing and semantic resources to arrive at a language-independent representation of meaning, thus focusing on the logical content of an information search rather than its form. This is especially significant for highly specialized domains such as medicine, on the other hand this approach presupposes the existence of large domain-specific thesauri.

The identification of terms and their mapping to concepts is the first stage of semantic analysis and its efficiency largely depends on the quality of linguistic processing on the one hand and the quality and coverage of the thesaurus on the other. Semantic relations between concepts represent another layer of information, which have the potential of making the document search even more detailed and specific, and possibly interactive by allowing the user to control the directions in which a query is expanded.

We report on a series of experiments performed to test and evaluate the role of semantic relations in CLIR. The work we describe was performed within a project on the systematic comparison of concept-based and corpus-based methods in cross-language medical information retrieval. We use

the Unified Medical Language System (UMLS) as the primary semantic resource and a corpus of English and German medical abstracts for development and evaluation of methods and tools. The paper focuses on semantic relations, which are a crucial element of medical knowledge representation. The basis of our experiments are the semantic relations as specified in the UMLS Semantic Network, which we seek to modify and expand for CLIR purposes. We describe a method for selecting relevant relations from those proposed by UMLS and a method for extracting new instances of relations based on statistical and NLP techniques.

2 Semantic Annotation for Concept-Based CLIR

2.1 UMLS and Semantic Relations

The essential part of any concept-based CLIR system is the identification of terms and their mapping to a language-independent conceptual level. Our basic resource for semantic annotation is UMLS, which is organized in three parts.

The **Specialist Lexicon** provides lexical information: a listing of word forms and their lemmas, part-of-speech and morphological information.

Second, the **Metathesaurus** is the core vocabulary component, which unites several medical thesauri and classifications into a complex database of concepts covering terms from 9 languages. Each term is assigned a unique string identifier, which is then mapped to a unique concept identifier (CUI). For example, the entry for the term *HIV pneumonia* in the Metathesaurus main term bank (MRCON) includes its CUI, language identifier, term status and finally the term string :

C0744975|ENG|P||HIV pneumonia|

The UMLS 2001 version includes 1.7 million terms mapped to 797,359 concepts, of which 1.4 million entries are English and only 66,381 German. Only the MeSH (Medical Subject Heading) part of the Metathesaurus covers both German and English, therefore we only use MeSH terms (564,011 term entries for English and 49,256 for German) for corpus annotation.

The third part is the **Semantic Network**, which provides a grouping of concepts according to their meaning into 134 semantic types (TUI). The concept above would be assigned to the class *T047, Disease or Syndrome*. The Semantic Network then specifies potential relations between those semantic types. There are 54 hierarchically organized domain-specific relations, such as *affects*, *causes*, *location_of* etc. All of them are binary relations (A is related to B).

2.2 Linguistic and Semantic Annotation

The foundation of our CLIR setting is the automatic linguistic and semantic annotation of the document collection, in our case a parallel corpus of about 9000 English and German medical abstracts, obtained from the Springer web site¹. For linguistic processing we are using ShProT, a shallow processing tool that consists of four integrated components: the SPPC tokenizer (Piskorski and Neumann, 2000), TnT (Brants, 2000) for part-of-speech tagging, Mmorph (Petitpierre and Russell, 1995) for morphological analysis and Chunkie (Skut and Brants, 1998) for phrase recognition.

The next stage is the annotation of various semantic information. At the level of terms, the following information is used:

- Concept Unique Identifier (CUI)
- Type Unique Identifier (TUI)
- Medical Subject Headings ID - an alternative code to the CUIs
- Preferred Term - a term that is marked as the preferred name for a particular concept

The identification of UMLS terms in the documents is based on morphological processing of both the term bank and the document, so that term lemmas are matched rather than word forms. The annotation tool matches terms of lengths 1 to 3 tokens, based on lemmas if available and word forms otherwise. Term matching on the sub-token level is also implemented to ensure the identification of terms that are a part of a more complex compound, which is crucial for German.

¹<http://link.springer.de>

In addition to concept identifiers (CUIs) we also annotate the codes of MeSH tree nodes. The decision to do so was based on our observation that the UMLS Semantic Network, especially the semantic types and relations, does not always adequately represent the domain-specific relationships. MeSH on the other hand has a transparent tree structure, from which both the semantic class of a concept and its depth in the tree can be inferred. For example, the terms *infarction* (C23.550.717.489) and *myocardial infarction* (C14.907.553.470.500) both belong to the group of diseases, but the node of the first term lies higher in the hierarchy as its code has fewer fields.

The term inventory in our documents is further expanded by integrating newly extracted terms provided by our project partners (cf. (Gaussier, 1998)). This slightly improves term-based retrieval, but since the new terms cannot be assigned a semantic type nor a MeSH code, they have no effect upon semantic relations.

Semantic relations are annotated on the basis of the UMLS Semantic Network, which defines binary relations between semantic types (TUIs) in the form of triplets, for example *T195 - T151 - T042* meaning *Antibiotic - affects - Organ or Tissue Function*. We search for all pairs of semantic types that co-occur within a sentence, which means that we can only annotate relations between items that were previously identified as UMLS terms. According to the Semantic Network relations can be ambiguous, meaning that two concepts may be related in several ways. For example:

Diagnostic Procedure assesses_effect_of Antibiotic	
Diagnostic Procedure analyzes	Antibiotic
Diagnostic Procedure measures	Antibiotic
Diagnostic Procedure uses	Antibiotic

Since the semantic types are rather general (e.g. *Pharmacological Substance*, *Patient or Group*), the relations are often found to be vague or even incorrect when they are mapped to a document. If for example the Semantic Network defines the relation *Therapeutic Procedure – method_of – Occupation or Discipline*, this may not hold true for all combinations of members of those two semantic classes, as seen in **discectomy – method_of –*

history. Given the ambiguity of relations and their generic nature, the number of potential relations found in a sentence can be high, which makes their usefulness questionable. A manual evaluation of automatic relation tagging in a small sample by medical experts showed that only about 17% of relations were correct, of which only 38% were perceived as significant in the context of information retrieval.

On the other hand, many relations undoubtedly present in our texts are not identified by automatic relation tagging. One possible reason for this may be the incompleteness of the Semantic Network, but a more accurate explanation is that relationships are constantly being woven between concepts occurring together in a specific context, thus creating novel or unexpected links that would not exist between concepts in isolation.

For the above reasons we developed methods to deal with each of the problems described, relation filtering and relation extraction.

3 Extending Existing Resources: Relation Filtering and Relation Extraction

3.1 Relation Filtering

The first task was to tackle relation ambiguity, i.e. to select correct and significant relations from the ones proposed by automatic UMLS lookup; a procedure we refer to as *relation filtering*. The method is composed of two steps following two initial hypotheses:

- Interesting relations will occur between interesting concepts.
- Relations are expressed by typical lexical markers, such as verbs.

3.1.1 Relation Filtering with IDF

Following our first hypothesis we expect interesting and true relations to occur between items that are specific rather than general, and thus not too frequent. To measure this specificity we use the *inverse document frequency (IDF)* of the concept's code (CUI), which assigns a higher weight to concepts occurring only in a subset of documents in the collection. We thus take N_t to be the

number of documents containing the CUI t and N the number of all documents.

$$IDF_t = \log_2 \frac{N}{N_t}$$

We decided to use IDF instead of the generally used TF-IDF, because term frequency (TF), if multiplied with IDF, will assign a higher score to frequent terms like *patient*, *therapy*, *disease*. Relations between items with the IDF weight below a certain value are removed; the threshold value was set experimentally to 2.7.

Consider the following example where two instances of the relation diagnoses are found in a sentence:

Diagnostic – diagnoses – Disease
Diagnostic – diagnoses – Lyme Disease

As the IDF weights of both *Diagnostic* and *Disease* are below the set threshold the first relation instance is removed.

3.1.2 Relation Filtering with Verbal Markers

Relations that are semantic links between (mainly) nominal items may be represented by various linguistic means or *lexical markers*. In a rule-based approach such markers would be specified manually, however we chose to use a co-occurrence matrix of *lexical verbs* and automatically tagged relations. This is based on the assumption that some verbs are more likely to signify a certain relation than others. The co-occurrences are normalized and non-lexical verbs filtered out, so that for each lexical verb we get a list of relations it most likely occurs with. This information is then used to remove relations that occur with an untypical verb.

Below are the frequencies of five relations that are assigned to the verb *activate*:

interacts_with (197)
produces (83)
affects (52)
disrupts (32)
result_of (29)

Table 1 shows the number of relation instances using UMLS [*umls*] and after each filtering step [*umls_idf_filt*, *umls_idf_vb_filt*].

3.2 Extraction of New Relation Instances

The identification of new instances of relations was based on observed co-occurrences of concepts, where instead of the semantic types (TUI) from the Metathesaurus we use MeSH classes. This gives us flexibility in choosing the number of semantic classes, depending on the level in the hierarchy.

The MeSH tree is organized into 15 top tree nodes, each of which is marked with a letter and subdivided into further branches. These top nodes are the following:

Anatomy [A]
Organisms [B]
Diseases [C]
Chemicals and Drugs [D]
Analytical, Diagnostic and Therapeutic Techniques and Equipment [E]
Psychiatry and Psychology [F]
Biological Sciences [G]
Physical Sciences [H]
Anthropology, Education, Sociology and Social Phenomena [I]
Technology and Food and Beverages [J]
Humanities [K]
Information Science [L]
Persons [M]
Health Care [N]
Geographic Locations [Z]

We use co-occurrences on the second level, meaning that we strip full MeSH codes assigned to each concept to only the top node letter and first order children. Looking at the structure of MeSH, this leaves us with 114 semantic classes, though some of them do not occur in our corpus. An example of a co-occurring MeSH pair is D3 [Heterocyclic Compounds] + C10 [Nervous System Diseases].

For each UMLS semantic relation we then compute a list of typical MeSH pairs, for example *treats*: D27|C23, D3|C23, E7|C23, E7|C2, Once these patterns of correspondence between pairs and relations are established, we may extract new instances of relations on the basis of co-occurring MeSH codes within the sentence.

Since the Semantic Network defines as many

as 54 relations, of which several are very generic (e.g. *associated_with*) and some very rare, we chose to limit the extraction procedure to 15 most frequent and at the same time most specific relations. These are: *result_of*, *location_of*, *interacts_with*, *produces*, *degree_of*, *issue_in*, *uses*, *performs*, *treats*, *measures*, *causes*, *disrupts*, *diagnoses*, *analyzes*.

Table 1 shows the number of relation instances in the corpus if we use only UMLS [*umls*], if we add new ones to those that may be found in UMLS [*umls_new*], the number of instances if we first perform the filtering and then add new ones [*umls_idf_vb_new*], and the number of instances we find using only our extraction method [*only_new*].

Corpus version	Relation instances
umls	702,449
umls_idf_filt	405,844
umls_idf_vb_filt	290,250
umls_idf_vb_new	461,823
umls_new	819,202
Only_new	1,009,847

Table 1: Number of relation instances in the corpus using UMLS, filtering and relation extraction

The extraction method can be tuned in terms of precision and recall by setting the MeSH-pair frequency threshold. For our current document collection and CLIR purposes this was set to 150, however other applications utilizing relation extraction, such as ontology building, might require a higher threshold.

4 Evaluation

The main goal of the experiments that we describe was to evaluate the usefulness of semantic relations in CLIR, where we explore the possibilities of modifying and expanding existing semantic resources, i.e. UMLS. The baseline of retrieval experiments is therefore to use UMLS as it is, and then compare performance achieved with pruning and expansion techniques.

To retrieve documents from the collection we are using a set of 25 medical queries, for which relevance assessments were provided by medical

experts. Those queries are available in English and German, and for the majority of previous CLIR evaluation tasks performed within our project we used German queries over the English document collection, in accordance with the envisaged user requirements. Unfortunately, due to low term coverage for German, only very few semantic relations were found on the query side, and it was therefore impossible to assess their value. For this reason we opted for using English queries over the English document collection, however without indexing tokens and lemmas but relying solely on semantic information. We believe that this – though still monolingual – setting allows us to generalise our observations for CLIR, because we are using concepts and relations as the interlingua.

All experiments were carried out using the Rondo² retrieval system, which indexes all semantic information provided in the XML annotated documents as separate categories: UMLS terms, MeSH terms, XRCE terms, semantic relations. The system uses a straight *lnu.ltn* weighting scheme. In the tables below we present the retrieval results in four columns: mean average precision (**mAP**), absolute number of relevant documents retrieved (**RD**), average precision at 0.1 recall (**AP01**) and precision for the top 10 documents retrieved (**P10**) (These metrics are also used in TREC experiments; cf. (Gaussier et al., 1998)). The total number of relevant documents for the 25 queries is 956.

4.1 Evaluation of relation filtering and relation extraction

Previous experiments within our project have shown that on the level of concepts MeSH codes achieve a higher precision than CUI’s from the UMLS (Volk et al., 2002). We therefore choose MeSH codes as the primary semantic category on the level of concepts (mesh), and to this we wish to compare retrieval results achieved by using semantic relations together with MeSH codes (mesh_semrel) as well as the results of using semantic relations only (semrel).

Table 2 gives the results obtained by using UMLS-based semantic annotations, to be consid-

²A retrieval system from Eurospider Information Technology AG

	mAP	RD	AP01	P10
umls_mesh	0.311	541	0.659	0.536
umls_mesh_semrel	0.302	542	0.644	0.544
umls_semrel	0.146	253	0.384	0.340

Table 2: Results of using UMLS-based semantic annotations

	mAP	RD	AP01	P10
umls_idf_filt	0.309	541	0.651	0.540
umls_idf_vb_filt	0.306	541	0.661	0.524
umls_idf_vb_new	0.305	541	0.665	0.520
umls_new	0.300	542	0.647	0.532
only_new	0.298	543	0.637	0.508

Table 3: Results of relation filtering and extraction indexing MeSH concepts and relations

ered our baseline. We see that the average precision of using only concepts (mAP = 0.311) decreases slightly if we introduce semantic relations, however with an equally slight increase in recall and precision at top 10 documents. Semantic relations are always based on prior identification of two concepts, they are thus very specific and inevitably produce low recall if used alone (mAP = 0.146). We nevertheless consider this information useful for assessing the impact of relation filtering and expansion.

To this baseline we now compare five versions of our document collection, each annotated with a different set of semantic relations. The first contains UMLS-based relations filtered with the IDF method (umls_idf_filt), the second was additionally filtered with the verb method (umls_idf_vb_filt). We then introduce newly extracted relation instances, first to the filtered version of the corpus (umls_idf_vb_new), then to the baseline UMLS-annotated version (umls_new) and finally, we annotate relations using only our method for extracting new relation instances (only_new). For each corpus version we use queries that were processed identically to the document collection.

Table 3 gives the results for the combination of MeSH codes and semantic relations, and Table 4 shows the results for semantic relations only.

	mAP	RD	AP01	P10
umls_idf_filt	0.126	203	0.315	0.280
umls_idf_vb_filt	0.107	175	0.282	0.264
umls_idf_vb_new	0.124	197	0.336	0.308
umls_new	0.153	259	0.419	0.344
only_new	0.116	213	0.363	0.280

Table 4: Results of relation filtering and extraction indexing relations only

If we use semantic relations on top of MeSH concept codes, almost no difference can be observed, except perhaps that filtering with IDF seems to have a positive effect on high-end precision and that adding new relations slightly increases recall. However if we look at the results obtained by using only semantic relations, the differences between approaches become more apparent. It seems that each filtering step significantly decreases both recall and precision, while adding new relations – as we would expect – works well. The highest precision and recall were achieved with a combination of UMLS annotation and new relations, and this combination also outperforms the baseline.

4.2 Evaluation with manually annotated queries

Relations represent a secondary, highly specific level of semantic information, which is difficult to evaluate in traditional CLIR settings. Within our project, responding to user requirements of the medical domain, we designed a retrieval prototype where semantic information can be used interactively. If semantic relations are understood as a specific point of view on top of the initial request, the user may first submit a query and then select the relations she would find useful.

In an approximation of this scenario we had our 25 queries first automatically tagged for terms and concepts, and then manually annotated for semantic relations by a medical expert. The expert was asked to use only the 15 relations listed above. Table 5 shows the retrieval results using manually annotated queries over all five versions of our corpus, where only semantic relations were indexed.

Although the overall results of this run are very

	mAP	RD	AP01	P10
umls_idf_filt	0.035	85	0.080	0.080
umls_idf_vb_filt	0.027	68	0.077	0.080
umls_idf_vb_new	0.031	77	0.080	0.080
umls_new	0.045	104	0.106	0.124
only_new	0.085	154	0.274	0.232

Table 5: Retrieval results using manually annotated queries (indexing only semantic relations)

low, which is due to the fact that manual annotation was much less ‘generous’ than the automatic, we see a dramatic increase in recall and precision using the corpus annotated only by our method. This indicates a high correspondence between the ‘true’, expert-provided information and the automatic extraction model, and thus confirms our intuitions about the relevance of MeSH co-occurrences.

5 Discussion

Although the initial motivation for this research was to enhance document retrieval by introducing semantic relations, results obtained from the set of experiments we describe above lead to other – possibly even more promising – fields of application. In a domain such as medicine where extensive semantic resources can be used for concept-based retrieval, the most influential factor in the performance of a CLIR system is concept or term coverage, while semantic relations should probably be implemented in an interactive way allowing the user to narrow the focus of an otherwise overproductive query. Another implication of the results presented is that a smaller set of relations is beneficial both for document retrieval and automatic extraction of relation instances.

In a broader context or in another domain these methods might be adapted to ontology expansion or, possibly in combination with term extraction, to ontology construction. Hand-crafted ontologies more often than not focus on concepts and hierarchical relations between them. Automatic relation extraction is an important method of revealing domain-specific, possibly even previously unknown links between concepts and is therefore an integral part of Text Mining and Knowledge Dis-

covery.

6 Related Work

Domain-specific multilingual thesauri have been used for English-German CLIR within social science (Gey and Jiang, 1999), while (Eichmann et al., 1998) describe the use of UMLS for French and Spanish queries on the OHSUMED text collection. Both of these approaches use the thesaurus to compile a bilingual lexicon, which is then used for query translation. Mandala et al. (Mandala et al., 1999) seek to complement WordNet by using corpus-derived thesauri and report improved performance in monolingual IR, however their approach only indirectly subsumes (unlabelled) relation extraction by using term co-occurrences.

Many approaches use lexical markers for extracting relations between terms or concepts (Hearst, 1992; Davidson et al., 1998; Finkelstein-Landau and Morin 1999), some also in combination with shallow parsing, but this method is generally low in recall and therefore not suitable for retrieval purposes. We are using lexical markers as probabilistic contexts for semantic classification, an approach similar to that of (Bisson et al., 2000). As for the relevance of MeSH classes for medical semantic relations, (Cimino and Barnett, 1993) already defined some combinations of top MeSH nodes that indicate specific medical relations. In our approach these combinations are established by statistical measures applied to our semantically annotated corpus, and we use a finer grained network of semantic classes.

7 Conclusions

In this paper we focus on the role of semantic relations, as specified in the UMLS Semantic Network, in concept-based medical CLIR. Proposed are two methods for improving their use: relation filtering and relation extraction. The evaluation of these methods shows that the first does not score well in retrieval, whereas relation extraction on the basis of co-occurrences of MeSH classes looks promising for query expansion. The evaluation with a set of manually annotated queries shows that newly extracted relation instances have

the highest level of correspondence with relations as identified by medical experts, which can especially be exploited in an interactive retrieval setting.

Future research will include learning semantic relations using classification techniques, where the context features of MeSH co-occurrences will be expanded from verbs to other linguistic markers including grammatical functions. For CLIR tasks it remains to be established which number of different relations works best. Although in our experiments relations do not lead to a major gain in precision and recall compared to using only concepts, the techniques we develop may find further application in related areas such as ontology construction and adaptation.

References

- G. Bisson, C. Nédellec, D. Canamero: Designing Clustering Methods for Ontology Building - The Mo'K Workbench. In: S. Staab, A. Maedche, C. Nédellec, P. WiemerHastings (eds.), Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI'00, Berlin, Germany, August 20-25, 2000.
- Brants T. 2000. TnT - A Statistical Part-of-Speech Tagger. In: *Proc. of the 6th ANLP Conference*, Seattle, WA.
- Carbonell J., Y. Yang, R. Frederking, R. D. Brown, Y. Geng, and D. Lee. 1997. Translingual Information Retrieval: A Comparative Evaluation. In: *Proc. of the Fifteenth International Joint Conference on Artificial Intelligence*.
- Cimino, J. and G. Barnett. Automatic knowledge acquisition from Medline. *Methods of Information in Medicine*, 32(2):120-130, 1993.
- Davidson, L., J. Kavanagh, K. Mackintosh, I. Meyer, and D. Skuce. 1998. Semi-automatic extraction of knowledge-rich contexts from corpora. *Proceedings, 1st Workshop on Computational Terminology (COMPUTERM'98)*, pages 50-56, Montreal.
- Eichmann D., M. Ruiz, and P. Srinivasan. 1998. Cross-Language Information Retrieval with the UMLS Metathesaurus. In: *Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia.
- M. Finkelstein-Landau and E. Morin. Extracting semantic relationships between terms: Supervised vs. unsupervised methods. In *International Workshop on Ontological Engineering on the Global Information Infrastructure*, pages 71-80, 1999.
- Gaussier E., G. Grefenstette, D. A. Hull, and B. M. Schulze. 1998. Xerox TREC-6 site report: Cross language text retrieval. In: *Proc. of the Sixth Text Retrieval Conference (TREC-6)*. National Institute of Standards Technology (NIST), Gaithersburg, MD.
- Gaussier, E. 1998. Flow Network Models for Word Alignment and Terminology Extraction from Bilingual Corpora. In: *Proceedings of the 36th Annual Meeting of the association for Computational Linguistics and the 17th International Conference on Computational Linguistics, COLING-ACL'98*, Montreal, Canada.
- Gey F. C., and H. Jiang. 1999. English-German Cross-Language Retrieval for the GIRT Collection - Exploiting a Multilingual Thesaurus. In: *Proc. of the Eighth Text REtrieval Conference (TREC-8)*, National Institute of Standards Technology (NIST), Gaithersburg, MD.
- Gonzalo J., F. Verdejo, and I. Chugur. 1999. Using EuroWordNet in a Concept-based Approach to Cross-Language Text Retrieval, *Applied Artificial Intelligence:13*, 1999.
- Hearst, M. Automatic Acquisition of Hyponyms from Large Text Corpora, *Proceedings of the Fourteenth International Conference on Computational Linguistics*, Nantes, France, July 1992.
- Hull D. A., and G. Grefenstette. 1996. Querying Across Languages: A Dictionary based Approach to Multilingual Information Retrieval. In: *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval: ACM SIGIR*. 1996. <http://www.xerox.fr/people/grenoble/hull/papers/sigir96.ps>
- Kraaij, W. and D. Hiemstra. 1998. TREC6 Working Notes: Baseline Tests for Cross Language Retrieval with the Twenty-One System. In: *TREC6 working notes*. National Institute of Standards and Technology (NIST), Gaithersburg, MD.
- Mandala, R., T. Tokunaga and H. Tanaka. 1999. Complementing WordNet with Roget's and Corpus-based Thesauri for Information Retrieval. In: *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, Bergen, Norway.
- Oard D. 1998. A comparative study of query and document translation for cross-lingual information retrieval In: *Proc. of AMTA*, Philadelphia, PA.

- Petitpierre D., and G. Russell. 1995. MMORPH - The Multext Morphology Program. *Multext deliverable report for the task 2.3.1*, ISSCO, University of Geneva, Switzerland.
- Piskorski, J. and G. Neumann. 2000. An intelligent text extraction and navigation system. In: *Proc. of the 6th RIAO*. Paris.
- Qui, Y. 1995. Automatic Query Expansion Based on a Similarity Thesaurus. *PhD thesis*, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.
- Skut W. and T. Brants. 1998. A Maximum Entropy partial parser for unrestricted text. In: *Proc. of the 6th ACL Workshop on Very Large Corpora (WVLC)*, Montreal, Canada.
- Volk M., B. Ripplinger, S. Vintar, P. Buitelaar, D. Raileanu, B. Sacaleanu. 2002. Semantic Annotation for Concept-Based Cross-Language Medical Information Retrieval. To appear in the International Journal of Medical Informatics, 2002.

Adaptive Text Extraction and Mining

Fabio Ciravegna
Department of Computer Science
University of Sheffield



F.Ciravegna@dcs.shef.ac.uk
www.dcs.shef.ac.uk/~fabio

Nicholas Kushmerick
Department of Computer Science
University College Dublin



nick@ucd.ie
www.cs.ucd.ie/staff/nick

What is IE

What can we extract from the Web and why?

- Introduction: (20 minutes)
 - what is IE
 - What can we extract from the Web
 - Why?
- Algorithms and methodologies (100 min)
- IE in practice (30 min)
- Conclusion, Future Work (10 min)
- Discussion

Ciravegna & Kushmerick: ECML-2003 Tutorial

The 'canonical' IE task

- **Input:**
 - Document
 - newspaper article, Web page, email message, ...
 - Pre-defined "information need"
 - frame slots, template fillers, database tuples, ...
- **Output**
 - The specific substrings/fragments of the document or labels that satisfy the stated information need, possibly organised in a template

- DARPA's 'Message Understanding Conferences/Competitions' since late 1980's; most recent: MUC-7, 1998.
- Recent interest in the machine learning and Web communities.

Ciravegna & Kushmerick: ECML-2003 Tutorial

IE Standard Tasks

- Preprocessing
 - Tokenization
 - Morphological Analysis
 - Part of Speech Tagging
- Information Identification
 - Named Entity Recognition
 - Template Filling (from the MUC)
 - Template Elements
 - Template Relations
 - Scenario Template

Ciravegna & Kushmerick: ECML-2003 Tutorial

NE Recognition & Coreference

19:16 **Moody's** rates Province of Saskatchewan A3

Moody's Investors Service Inc said it assigned an A3 rating to the Province of Saskatchewan's **CS\$115 million** bond offering that was priced today.

The sale is a reopening of the province's **9.6 percent** bonds due **February 4, 2022**. Proceeds will be used for **government purposes**, mainly Saskatchewan Power Corp.

Ciravegna & Kushmerick: ECML-2003 Tutorial

Template Filling

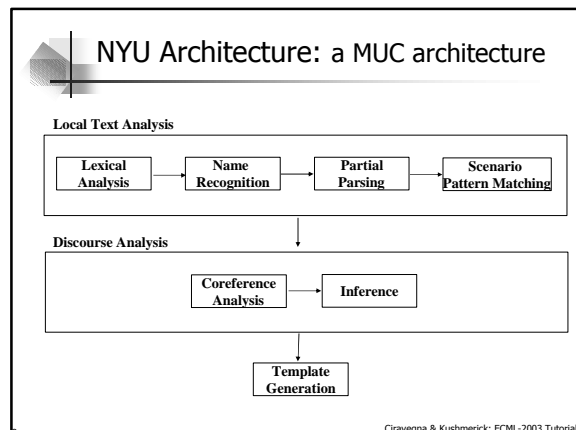
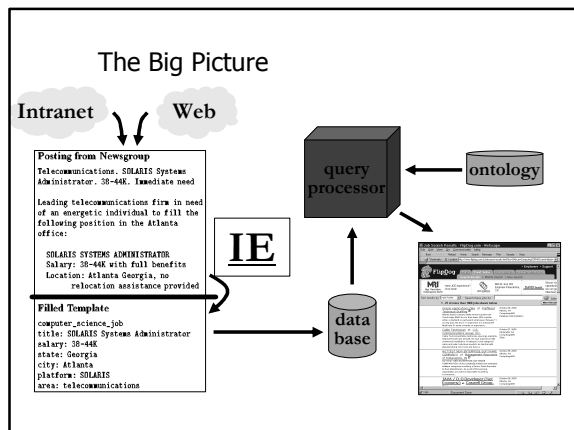
19:16 Moody's rates Province of Saskatchewan A3

Moody's Investors Service Inc said it assigned an A3 rating to the Province of Saskatchewan's **CS\$115 million** bond offering that was priced today.

The sale is a reopening of the province's **9.6 percent** bonds due **February 4, 2022**. Proceeds will be used for government purposes, mainly Saskatchewan Power Corp.

amount	CS\$115 million
issuer	Province of Saskatchewan
placement-date	today
maturity	February 4, 2022
rate	9.6 percent

Ciravegna & Kushmerick: ECML-2003 Tutorial



Semantic Web

- A brain for Human Kind
- From Information-based to Knowledge-Based
- Processable Knowledge means:
 - Better Retrieval
 - Reasoning
- Where can IE contribute?

Building the SW

- Document annotation
 - Manually associate documents (or parts) to ontological descriptions
 - Document classification for retrieval
 - Where can I buy an Hamster?
 - Pet shop web page -> pet shop concept -> hamster
 - Knowledge annotation
 - Where can I find a hotel in Berlin where single rooms cost less than 400€?
 - The Hotel is located in central Berlin and the cost for a single room is 300€
 - Editors are currently available for manual annotation of texts

IE for Annotating Documents

- Manual annotation is
 - Expensive
 - Error prone
- IE can be used for annotating documents
 - Automatically
 - Semi-Automatically
 - As user support
- Advantages
 - Speed
 - Low cost
 - Consistency
 - Can provide automatic annotation different from the one provided by the author(!)

SW for Knowledge Management

- SW is important for everyday Internet users
- SW is necessary for large companies
 - Millions of documents where knowledge is interspersed
 - Most documents are now
 - web-based
 - Available over an Intranet
 - Companies are valued for their
 - Tangible assets (e.g. plants)
 - Intangible assets (e.g. knowledge)
 - Knowledge is stored in
 - mind of employees
 - Documentation
 - Companies spend 7-10% of revenues for KM

Why Adaptive Systems?

- Writing IE systems by hand is difficult and error prone
 - Extraction languages can be quite complex
 - Tedious write-test-debug-rewrite cycle
- Adaptive systems learn from user annotations
 - the person tells the learning algorithm **what** to extract: The learner figures out **how**
- Advantages
 - Annotating text is simpler & faster than writing rules.
 - Domain independent
 - Domain experts don't need to be linguists or programmers.
 - Learning algorithms ensure full coverage of examples.

13

Cravegna & Kushmerick: ECML-2003 Tutorial

Algorithms and Methodologies

A dip into the details of IE for the Web

- Introduction: (20 minutes)
- Algorithms and methodologies (100 min)
 - Wrapper induction
 - Boosted wrapper induction
 - Hidden Markov models
 - Exploiting linguistic constraints
- IE in practice (30 min)
- Conclusion, Future Work (10 min)
- Discussion

14

Cravegna & Kushmerick: ECML-2003 Tutorial

Algorithms: Outline

- Wrappers
- Hand-coded wrappers
- Wrapper induction
- Learning highly expressive wrappers
- Boosted wrapper induction
- Hidden Markov models
- Exploiting linguistic constraints

structured data



natural text

15

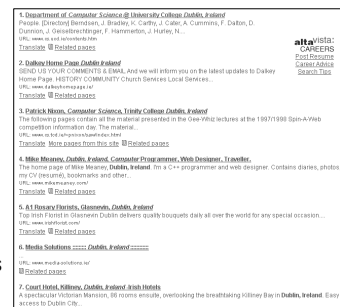
Cravegna & Kushmerick: ECML-2003 Tutorial

Wrapper induction

Highly regular source documents

Relatively simple extraction patterns

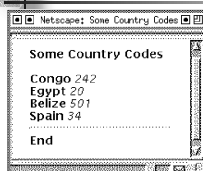
Efficient learning algorithms



16

Cravegna & Kushmerick: ECML-2003 Tutorial

Wrappers: Example and Toolkits



< (Congo, 242)
(Egypt, 20)
(Belize, 501)
(Spain, 34) >

Wrapper toolkits: Specialized programming environments for writing & debugging wrappers **by hand**

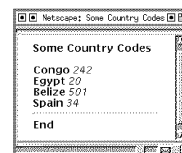
Examples

- World Wide Web Wrapper Factory[db.cis.upenn.edu/W4F]
- Java Extraction & Dissemination of Information [www.darmstadt.gmd.de/oasys/projects/jedi]

17

Cravegna & Kushmerick: ECML-2003 Tutorial

Wrappers: Delimiter-based extraction



```
<HTML><TITLE>Some Country Codes</TITLE>
<B>Congo</B> <I>242</I><BR>
<B>Egypt</B> <I>20</I><BR>
<B>Belize</B> <I>501</I><BR>
<B>Spain</B> <I>34</I><BR>
<BODY></HTML>
```

Use , , <I>, </I> for extraction

18

Cravegna & Kushmerick: ECML-2003 Tutorial

"Left-Right" wrappers

Left-Right wrapper $\equiv 2K$ strings

$\langle l_1, r_1, \dots, l_K, r_K \rangle$

left delimiters right delimiters

procedure ExtractCountryCodes
while there are more occurrences of
1. extract Country between and
2. extract Code between <I> and </I>

procedure ExtractAttributes:
while there are more occurrence of l_1
1. extract 1st attribute between l_1 and r_1
...
K. extract Kth attribute between l_K and r_K

[Kushmerick et al, IJCAI-97; Kushmerick AIJ-2000]

19

Cravegna & Kushmerick: ECML-2003 Tutorial

Wrapper induction

examples

hypothesis

Thai food is spicy.
Vietnamese food is spicy.
German food isn't spicy.

Asian food
is spicy.

wrapper

20

Cravegna & Kushmerick: ECML-2003 Tutorial

Learning LR wrappers

labeled pages

wrapper

$\langle l_1, r_1, \dots, l_K, r_K \rangle$

Example: Find 4 strings
, , <I>, </I>
(l_1, r_1, l_2, r_2)

21

Cravegna & Kushmerick: ECML-2003 Tutorial

LR: Finding r_1

<HTML><TITLE>Some Country Codes</TITLE>
Congo <I>242</I>

Egypt <I>20</I>

Belize <I>501</I>

Spain <I>34</I>

</BODY></HTML>

r_1 can be any *prefix*
eg

22

Cravegna & Kushmerick: ECML-2003 Tutorial

LR: Finding l_1, l_2 and r_2

l_1 can be any *suffix*
eg
 l_2 can be any *suffix*
eg <I>
 r_2 can be any *prefix*
eg </I>

23

Cravegna & Kushmerick: ECML-2003 Tutorial

A problem with LR wrappers

Distracting text in head and tail

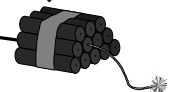
<HTML><TITLE>Some Country Codes</TITLE>
<BODY>Some Country Codes<P>
Congo <I>242</I>

Egypt <I>20</I>

Belize <I>501</I>

Spain <I>34</I>

<HR>End</BODY></HTML>



24

Cravegna & Kushmerick: ECML-2003 Tutorial

One (of many) solutions: HLRT

Ignore page's **head** and **tail**

```
<HTML><TITLE>Some Country Codes</TITLE>
<BODY><B>Some Country Codes</B><P>
<B>Congo</B> <I>242</I><BR>
<B>Egypt</B> <I>20</I><BR>
<B>Belize</B> <I>501</I><BR>
<B>Spain</B> <I>34</I><BR>
<HR><B>End</B></BODY></HTML>
```

end of head
body
start of tail

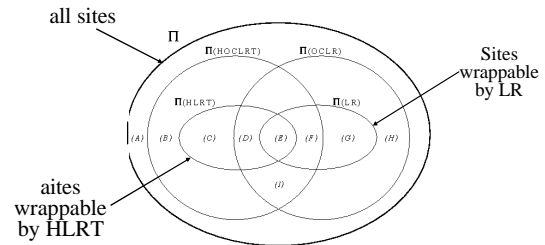
Head-**L**eft-**R**ight-**T**ail wrappers

25

Cravegna & Kushmerick: ECML-2003 Tutorial

Expressiveness

Theorem:



26

Cravegna & Kushmerick: ECML-2003 Tutorial

Coverage

wrapper class(es)	coverage (%)
$LR \cup HLRT \cup OCLR \cup HOCLRT \cup N-LR \cup N-HLRT$	70
$LR \cup HLRT \cup OCLR \cup HOCLRT$	60
$LR \cup OCLR$	53
LR	53
$OCLR$	53
$HLRT \cup HOCLRT$	57
$HLRT$	57
$HOCLRT$	57
$N-LR \cup N-HLRT$	53
$N-LR$	13
$N-HLRT$	50
$N-LR \cup N-HLRT$ but not $LR \cup HLRT \cup OCLR \cup HOCLRT$	25

Fraction of randomly-selected "data-heavy" Web sites (search engines, retail, weather, news, finance, ...) for which wrapper in a given class was learned.

27

Cravegna & Kushmerick: ECML-2003 Tutorial

Sample complexity

- The key problem with machine learning: Training data is expensive and tedious to generate
- In practice, active learning and specialized algorithms have reduced training requirements considerably
- But this isn't theoretically satisfying
- Computational Learning Theory:
 - Time complexity:** Time required by an algorithm to terminate, as a function of problem parameters
 - Sample complexity:** Training data required by a learning algorithm to converge to correct hypothesis, as a function of problem parameters

28

Cravegna & Kushmerick: ECML-2003 Tutorial

A Model of Sample Complexity

$P[\text{correct wrapper}] = f(\text{size of documents, number of documents, number of attributes per record})$

Analyze wrapper learning task to derive this function

(Actually, we can compute only a bound on this probability.)

Just like time/space complexity:

$\text{time}[\text{learn wrapper}] = g(\text{size of documents, number of documents, number of attributes per record})$

29

Cravegna & Kushmerick: ECML-2003 Tutorial

PAC results - LR wrappers

Theorem: Suppose we learn LR wrapper W from training set E , where the longest document has length R and each record contains K attributes. If

$$|E| \geq \frac{1}{\epsilon} (2K \ln R - \ln \delta)$$

then W is probably approximately correct

$\text{error}(W) < \epsilon$

with probability at least $1 - \delta$

30

Cravegna & Kushmerick: ECML-2003 Tutorial

More sophisticated wrappers

- LR & HLRT wrappers are extremely simple (though useful for $\sim 2/3$ of real Web sites!)
- Recent wrapper induction research has explored...
 - **more expressive wrapper classes**
[Muslea et al, Agents-98; Hsu et al, JIS-98; Thomas et al, JIS-00, ...]
 - Disjunctive delimiters
 - Sequential/landmark-based delimiters
 - Multiple attribute orderings
 - Missing attributes
 - Multiple-valued attributes
 - Hierarchically nested data
 - **Wrapper verification/maintenance**
[Kushmerick, AAAI-1999; Kushmerick WWWJ-00; Cohen, AAAI-1999; Minton et al, AAAI-00]

81

Cravegna & Kushmerick: ECML-2003 Tutorial

One of my favorites

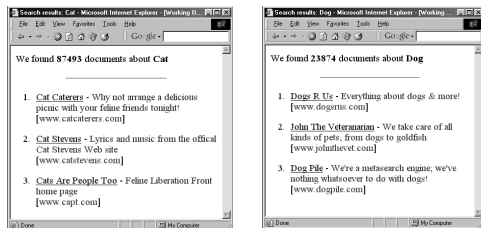
- **Roadrunner**
[Valter Crescenzi et al; Univ Roma 3]
- **Unsupervised wrapper induction**
 - They research databases, not machine learning, so they didn't realize training data was needed :-)
- **Intuition:**
 - Pose two different queries
 - The common bits of the documents come from the template and can be ignored
 - The bits that are different are the data that we're looking for

82

Cravegna & Kushmerick: ECML-2003 Tutorial

Roadrunner - Example

- Common content = Part of template
- Varying content = The data!



- Complications: Dynamic but unwanted content -- eg advertisements or timestamps

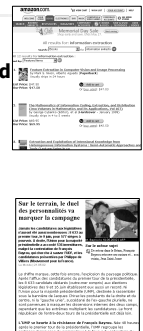
83

Cravegna & Kushmerick: ECML-2003 Tutorial

Algorithms: Outline

- ✓ Wrappers
- ✓ Hand-coded wrappers
- ✓ Wrapper induction
- ✓ Learning highly expressive wrappers
- Boosted wrapper induction
- Hidden Markov models
- Exploiting linguistic constraints

structured data
↑
natural text



84

Cravegna & Kushmerick: ECML-2003 Tutorial

Boosted wrapper induction

[Freitag & Kushmerick, AAAI-00]

- Wrapper induction is suitable only for rigidly-structured machine-generated HTML...
- ... or is it?!
- Can we use simple patterns to extract from natural language documents?

... Name: Dr. Jeffrey D. Hermes ...
... Who: Professor Manfred Paul ...
... will be given by Dr. R. J. Pangborn ...
... Ms. Scott will be speaking ...
... Karen Shriver, Dept. of ...
... Maria Klawe, University of ...

85

Cravegna & Kushmerick: ECML-2003 Tutorial

BWI: The basic idea

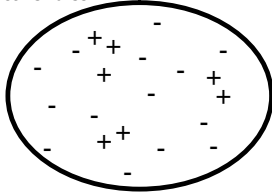
- Learn "wrapper-like" patterns for natural texts
pattern = exact token sequence
- Learn many such "weak" patterns
- Combine with boosting to build "strong" ensemble pattern
- Of course, not all natural text is sufficiently regular!
- Demo: www.smi.ucd.ie/bwi

86

Cravegna & Kushmerick: ECML-2003 Tutorial

Covering Algorithms

- Generalization of Covering Algorithm for learning disjunctive rules

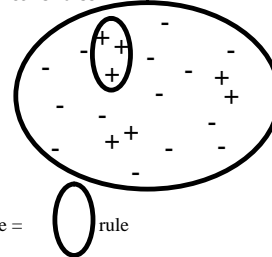


87

Cravegna & Kushmerick: ECML-2003 Tutorial

Covering Algorithms

- Generalization of Covering Algorithm for learning disjunctive rules

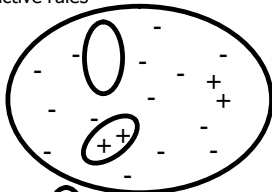


88

Cravegna & Kushmerick: ECML-2003 Tutorial

Covering Algorithms

- Generalization of Covering Algorithm for learning disjunctive rules



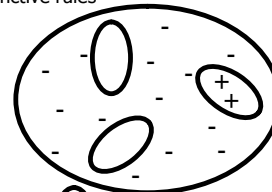
Learned Rule = rule or rule

89

Cravegna & Kushmerick: ECML-2003 Tutorial

Covering Algorithms

- Generalization of Covering Algorithm for learning disjunctive rules



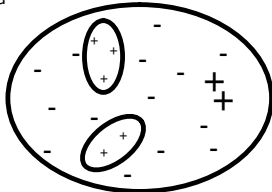
Learned Rule = rule or rule or rule

90

Cravegna & Kushmerick: ECML-2003 Tutorial

Boosting = Generalized Covering

- When learn rules on iteration t , give less weight to (but don't entirely discard) training examples successfully handled in iterations $1, 2, \dots, t-1$
- Equivalently: Give more weight to training data that has not yet been covered



91

Cravegna & Kushmerick: ECML-2003 Tutorial

Boosting [Schapire & Singer, 1998]

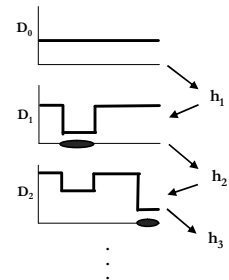
$D_1(i)$ = uniform distribution over training examples

for $t = 1, \dots, T$

train: use distribution D_t to learn weak hypothesis:
 $h_t: X \rightarrow R$

reweight: choose a_t and modify distribution D_t to emphasize examples missed by h_t :
 $D_{t+1}(i) = D_t(i) \exp(-a_t y_t h_t(x_i))$

return:
 $H(x) = \text{sign}(\sum a_t h_t(x))$

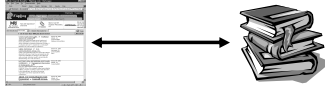


92

Cravegna & Kushmerick: ECML-2003 Tutorial

Boosted Wrapper Induction: Controversial(?) Conclusion

- Is the **Great Web -vs- Natural Text Chasm** more apparent than real?



- IE is possible if the documents contain regularities that can be exploited
- But the "reason" (eg, linguistic -vs- markup) for these regularities doesn't much matter
- See also Soderland's WHISK & Webfoot

60

Cravegna & Kushmerick: ECML-2003 Tutorial

Algorithms: Outline

- ✓ Wrappers
- ✓ Hand-coded wrappers
- ✓ Wrapper induction
- ✓ Learning highly expressive wrappers
- ✓ Boosted wrapper induction
- Hidden Markov models
- Exploiting linguistic constraints

structured data
↑
natural text



61

Cravegna & Kushmerick: ECML-2003 Tutorial

Hidden Markov models

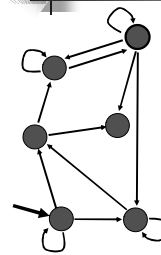
- Previous discussion examine systems that use explicit extraction patterns/rules
- HMMs are a powerful alternative based on statistical token models rather than explicit extraction patterns.

[Leek, UC San Diego, 1997; Bikel et al, ANLP-97, MLJ 99; Freitag & McCallum, AAAI-99 MLIE Workshop; Seymore, McCallum & Rosenfeld, AAAI-99 MLIE Workshop; Freitag & McCallum, AAAI-2000]

62

Cravegna & Kushmerick: ECML-2003 Tutorial

HMM formalism



HMM = states s_1, s_2, \dots
special start state s_1
special end state s_n
token alphabet a_1, a_2, \dots
state transition probs $P(s_i | s_j)$
token emission probs $P(a_i | s_j)$

Widely used in many language processing tasks,
e.g. speech recognition [Lee, 1989], POS tagging [Kupiec, 1992], topic detection [Yamron et al, 1998]

63

Cravegna & Kushmerick: ECML-2003 Tutorial

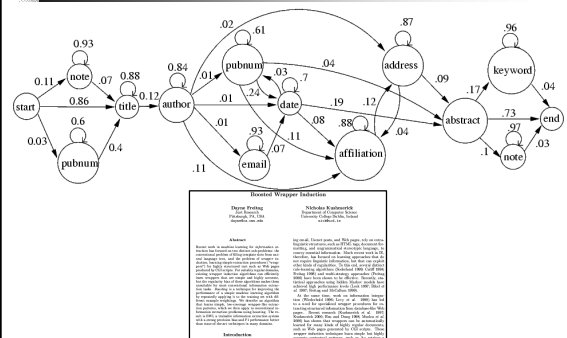
Applying HMMs to IE

- **Document** \Rightarrow generated by a stochastic process modelled by an HMM
- **Token** \Rightarrow word
- **State** \Rightarrow "reason/explanation" for a given token
 - 'Background' state emits tokens like 'the', 'said', ...
 - 'Money' state emits tokens like 'million', 'euro', ...
 - 'Organization' state emits tokens like 'university', 'company', ...
- **Extraction:** The Viterbi algorithm is a dynamic programming technique for efficiently computing the most likely sequence of states that generated a document.

64

Cravegna & Kushmerick: ECML-2003 Tutorial

HMM for research papers [Seymore et al, 99]



65

Cravegna & Kushmerick: ECML-2003 Tutorial

Learning HMMs

Good news:

- If training data tokens are tagged with their generating states, then simple frequency ratios are a maximum-likelihood estimate of transition/emission probabilities. (Use smoothing to avoid zero probs for emissions/transitions absent in the training data.)

Great news:

- Baum-Welch algorithm trains HMM using unlabelled training data!

Bad news:

- How many states should the HMM contain?
- How are transitions constrained?
- Insufficiently expressive \Rightarrow Unable to model important distinctions
- Overly-expressive \Rightarrow sparse training data, overfitting

55

Cravegna & Kushmerick: ECML-2003 Tutorial

HMM example

"Seminar announcements" task

```
<0.15.4.95.15.11.55.rudibear+@CMU.EDU.0>
Type: cmu.andrew.assocs.UEA
Topic: Re: entrepreneurship speaker
Dates: 17-Apr-95
Time: 7:00 PM
PostedBy: Colin S Osburn on 15-Apr-95 at 15:11 from CMU.EDU
Abstract:

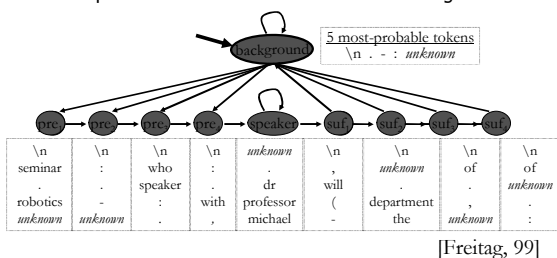
hello again
to reiterate
there will be a speaker on the law and startup business
this monday evening the 17th
it will be at 7pm in room 261 of GSIA in the new building, ie
upstairs.
please attend if you have any interest in starting your own
business or
are even curious.
Colin
```

56

Cravegna & Kushmerick: ECML-2003 Tutorial

HMM example, continued

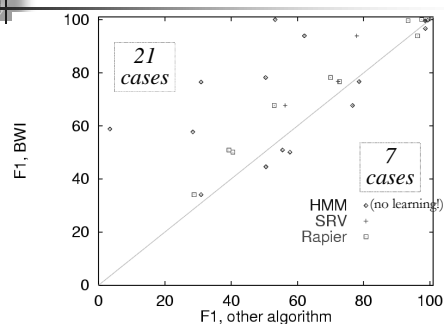
Fixed topology that captures limited context:
4 "prefix" states before & 4 "suffix" after target state



57

Cravegna & Kushmerick: ECML-2003 Tutorial

Evaluation

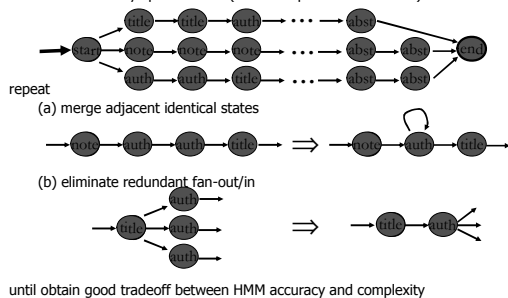


58

Cravegna & Kushmerick: ECML-2003 Tutorial

Learning HMM structure [Seymore et al, 1999]

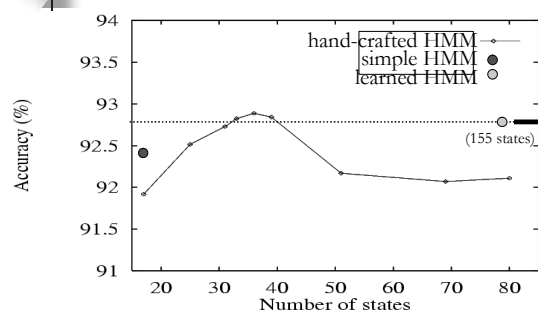
start with maximally-specific HMM (one state per observed word):



59

Cravegna & Kushmerick: ECML-2003 Tutorial

Evaluation



60

Cravegna & Kushmerick: ECML-2003 Tutorial

Algorithms: Outline

- ✓ Wrappers
- ✓ Hand-coded wrappers
- ✓ Wrapper induction
- ✓ Learning highly expressive wrappers
- ✓ Boosted wrapper induction
- ✓ Hidden Markov models
- Exploiting linguistic constraints

structured
data

↑

natural
text

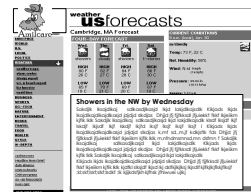


61

Cravegna & Kushmerick: ECML-2003 Tutorial

Exploiting linguistic constraints

- IE research has its roots in the NLP community
- many extraction tasks require non-trivial linguistic processing
- Web Documents types can range from free texts to rigid HTML documents (e.g. tables)
 - Even a mixture of them!
- Is NLP robust enough to cope with such situations?



62

Cravegna & Kushmerick: ECML-2003 Tutorial

Current Approaches

- NLP Approaches (MUC-like Approaches)
 - Ineffective on most Web-related texts:
 - web pages/emails
 - stereotypical but ungrammatical texts
 - Extra-linguistic structures convey information
 - HTML tags, Document formatting, Regular stereotypical language
- Wrapper induction systems
 - Designed for rigidly structured HTML texts
 - Ineffective on unstructured texts
 - Approaches avoid generalization over flat word sequence
 - Data Sparseness on free texts

63

Cravegna & Kushmerick: ECML-2003 Tutorial

Lazy NLP based Algorithm

- Learns the best level of language analysis for a specific IE task mixing deep linguistic and shallow strategies
 1. Initial rules: shallow wrapper-like rules
 2. Linguistic Information (LI) progressively added to rules
 3. Addition stopped when LI becomes
 - unreliable
 - ineffective
- Lazy NLP learns best strategy for each information/context separately
 - Example:
 - Using parsing for recognising the speaker in seminar announcements,
 - Using shallow approaches to spot the seminar location

64

Cravegna & Kushmerick: ECML-2003 Tutorial

(LP)²

[Cravegna 2001 – IJCAI 01- ATEM01]

- Covering algorithm based on LazyNlp
- Single tag learning (e.g. </speaker>)
- Tagging Rules
 - Insert annotation in texts
- Correction Rules
 - Correct imprecision in information identification by shifting tags to the correct position

TBL-like, with some fundamental differences

65

Cravegna & Kushmerick: ECML-2003 Tutorial

Tagging and Correction Rules: examples

the seminar at <time> 4 pm </time> will

Condition on Words	Action: Insert Tag
the	
seminar	
at	<time>
4	
pm	

Initial rules= window of conditions on words

The seminar at 4 </time> PM will be held in Room 201

Condition	Action
word	wrong tag
at	
4	</time>
pm	</time>

Cravegna & Kushmerick: ECML-2003 Tutorial

Rule Generalisation

- Each instance is generalised by reducing its pattern in length
- Generalizations are tested on training corpus
- Best k rules generated from each instance reporting:
 - Smallest error rate (wrong/matches)
 - Greatest number of matches
 - Cover different examples
- Conditions on words are replaced by information from NLP modules
 - Capitalisation
 - Morphological analysis
 - Generalizes over gender/number
 - POS tagging
 - Generalizes over lexical categories
 - User-defined dictionary or gazetteer
 - Named Entity Recognizer

Implemented as a general to specific beam search with pruning (AQ-like)

Cravegna & Kushmerick: ECML-2003 Tutorial

Example of generalization

the seminar at <time> 4 pm will

Condition		Additional Knowledge			Action
Word	Lemma	LexCat	Case	SemCat	Tag
the	the	det	low		
seminar	seminar	noun	low		
at	at	prep	low		
4		digit	low		<time>
pm	noun	low		timeid	
will	will	verb	low		

Condition					Action
Word	Lemma	LexCat	Case	SemCat	Tag
	at				
		digit			<time>
				timeid	

Details of the algorithm in [Cravegna 2001 - ATEM01] Cravegna & Kushmerick: ECML-2003 Tutorial

CMU: detailed results

	(LP) ²	BW1	HMM	SRV	Rapier	Whisk
speaker	77.6	67.7	76.6	56.3	53.0	18.3
location	75.0	76.7	78.6	72.3	72.7	66.4
stime	99.0	99.6	98.5	98.5	93.4	92.6
etime	95.5	93.9	62.1	77.9	96.2	86.0
All Slots	86.0	83.9	82.0	77.1	77.3	64.9

- Best overall accuracy
- Best result on speaker field
- No results below 75%

Cravegna & Kushmerick: ECML-2003 Tutorial

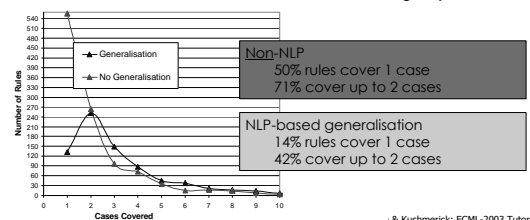
Effect of Generalization(1)

Effectiveness and reduction in data sparseness

Slot	(LP) ² _G	(LP) ² _{NG}
speaker	72.1	14.5
location	74.1	58.2
stime	100	97.4
etime	96.4	87.1
All slots	89.7	78.2

Most Interesting

With comparable effectiveness on training corpus!



Cravegna & Kushmerick: ECML-2003 Tutorial

Best level of Generalization

- ITC seminar announcements (mixed Italian/English)
 - Date, time, location generally in Italian
 - Speaker, title and abstract generally in English
- English POS also for the Italian part
- NLP-based outperforms other version

	Words	POS	NE
speaker	74.1	75.4	84.3
title	62.8	62.4	62.8
date	90.8	93.4	93.9
time	100	100	100
location	95.0	95.0	95.5

Cravegna & Kushmerick: ECML-2003 Tutorial

Linguistic constraints: Conclusions

- Linguistic phenomena can't be handled by simple wrapper-like extraction patterns
- Even shallow linguistic processing (eg POS tagging) can improve performance dramatically.
 - NOTE: linguistic processing must be regular, not necessarily correct!
 - Example
(LexCat:NNP +
 +
)<SPEAKER>(NER:<person>)
none of the covered 32 examples starts actually with an NNP
- What about more sophisticated NLP techniques?
 - Extension to parsing and coreference resolution?

Cravegna & Kushmerick: ECML-2003 Tutorial

Putting IE into Practice

Enabling non-experts to port IE systems

- Introduction: (20 minutes)
 - what is IE, what can we extract from the Web and why?
- Algorithms and methodologies (100 min)
- IE in practice (30 min)
 - The adaptation problem (20 min)
 - WEB + IE: examples of systems (10 min)
- Conclusion, Future Work (10 min)
- Discussion

Cravegna & Kushmerick: ECML-2003 Tutorial

Motivation

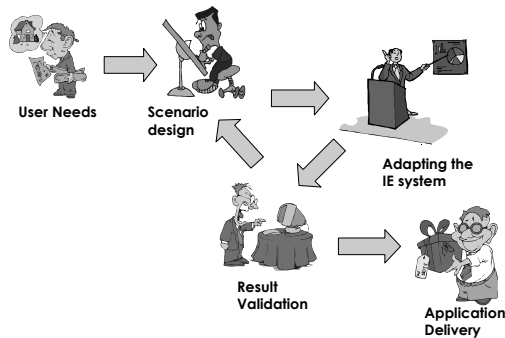
- Impact on the web community will come only if:
 - IE systems are portable by non IE experts
 - Low cost porting
- Non experts
 - Need specific easy to use tools to:
 - Design application
 - Tune application
 - Deliver application
 - Need support during the whole IE application definition process

In summarising the summary of the summary:
people are a problem.

Douglas Adams
The Restaurant at the End of the Universe

Cravegna & Kushmerick: ECML-2003 Tutorial

Application Development Cycle



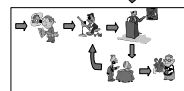
Scenario design

- Task: mapping user wishes into templates
- Necessity:
 - Supporting users in:
 - relevant information identification
 - scenario organization
- Relevant Information Identification:
 - Different situations:
 - User with developed scenario
 - System: no action, but...
 - User with preliminary scenario to be refined
 - System helps in refining
 - User with no scenario
 - System helps in
 - Identifying relevant information
 - Organising it into a scenario

Cravegna & Kushmerick: ECML-2003 Tutorial

Training

- User can select unrepresentative corpora
 - Unbalanced wrt genres
 - System validates corpus wrt a large corpus
 - Comparing formal features
 - Unwanted regularities (use of keywords for selection)
 - System looks for unusual regularities
 - Irrelevant texts (sensitive information)
 - No solution to stupidity



Cravegna & Kushmerick: ECML-2003 Tutorial

Tagging Corpora

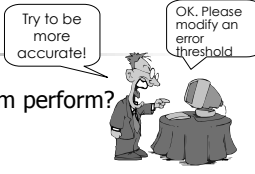
- Problems:
 - Tagging texts can:
 - Be difficult and boring
 - Take a long time
- Effect:
 - Mistakes in tagging
 - High cost
- System:
 - reduce/eliminate need for annotated data
 - **Bootstrapping**: from user-defined "seed examples" to system-retrieved similar examples
 - **Active learning**: selection of examples to annotate from unlabeled corpus

Helps in discovering
new relations

Helps in focusing on
unusual information shape


Cravegna & Kushmerick: ECML-2003 Tutorial

Result Validation



- How well does the system perform?
 - Solution:
 - Facilities for:
 - Inspecting tagged corpus
 - Showing details on correctness
 - Statistics on corpus
 - Details on errors (highlight correct/incorrect/missing) (e.g. MUC scorer is an excellent tool)
 - Influencing system behavior
 - Solution
 - Interface for bridging the user's qualitative vision and the system's numerical vision

Application Delivery



- Problem:
 - Incoming texts deviate from training data
 - Training corpus non representative
 - Document features change in time
- Solution:
 - Monitoring application.
 - Warn user if incoming texts' features are statistically different from training corpus:
 - Formal features: texts length, distribution of nouns
 - Semantic features: distribution of template fillers

Putting IE into Practice (2)

Some examples of Adaptive User-driven IE for real world applications

Learning Pinocchio

- Commercial tool for adaptive IE
 - Based on the (LP)² algorithm
 - Adaptable to new scenarios/applications by:
 - Corpus tagging via SGML
 - A user with analyst's knowledge
- Applications
 - "Tombstone" data from Resumes (Canadian company) (E)
 - IE from financial news (Kataweb) (I)
 - IE from classified ads (Kataweb) (I)
 - Information highlighting (intelligence)
 - (Many others I have lost track of...)
- A number of licenses released around the world for application development

[Ciravegna 2001 - IJCAI]
<http://tcc.itc.it/research/textec/tools-resources/learningpinocchio/>

Application development time


Resumes:

- Scenario definition: 10 person hours
- Tagging 250 texts: 14 person hours
- Rule induction: 72 hours on 450MHz computer
- Result validation: 4 hours

Contact:
 Alberto Lavelli
 ITC-Irst
lavelli@itc.it
<http://tcc.itc.it/research/textec/tools-resources/learningpinocchio/>

Amilcare

active annotation for the Semantic Web



Tool for adaptive IE from Web-related texts

- Based on (LP)²
- Uses Gate and Annie for preprocessing
- Effective on different text types
 - From free texts to rigid docs (XML, HTML, etc.)
- Integrated with
 - MnM (Open University) Ontomat (University of Karlsruhe)
 - Gate (U Sheffield)
- **Adapting Amilcare:**
 - Define a scenario (ontology)
 - Define a Corpus of documents
 - Annotate texts
 - Via MnM, Gate, Ontomat
 - Train the system
 - Tune the application (*)
 - Deliver the application

[Ciravegna 2002 - SIGIR]
www.dcs.shef.ac.uk/~fabio/Amilcare.html

Non-Intrusive Active Learning

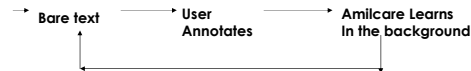
- Amilcare is specifically designed as companion for text annotation
 - It can be inserted in the usual tagging environment
 - It works in the background
 - At some point it will start helping the user in tagging



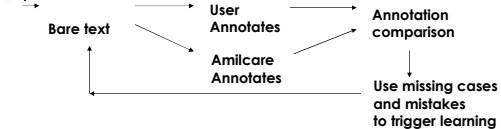
85

Cravegna & Kushmerick: ECML-2003 Tutorial

Bootstrapping Annotation



Learning to annotate

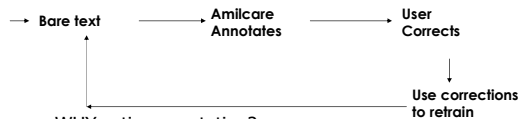


86

Cravegna & Kushmerick: ECML-2003 Tutorial

Active Annotation

- When Amilcare's rules reach a user-defined accuracy

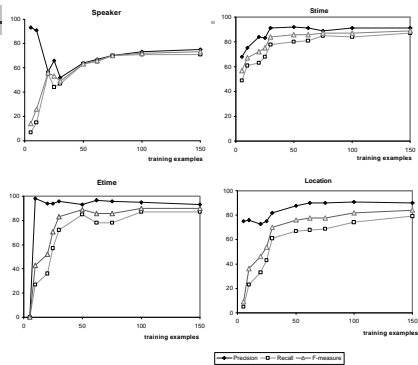


- WHY active annotation?
 - Focuses the slow and expensive user activity on uncovered cases
 - Avoids annotating covered cases
 - Validating extracted information is
 - Simpler & less error prone
- Than annotating bare texts speeding up the process of corpus annotation considerably.

87

Cravegna & Kushmerick: ECML-2003 Tutorial

Is IE useful as Help for Tagging?



88

IL-2003 Tutorial

Conclusions on IE and Tagging

Tag	Amount of Texts needed for training	Prec	Rec
stime	30	91	78
etime	20	96	72
location	30	82	61
speaker	100	75	70

- Integration of IE (Amilcare+Gate) and Ontology-based Annotation Tools (MnM and Ontomat)
- First step towards a new generation of OEs
- Active Learning can provide an interesting interaction modality
 - User friendly
 - Adaptable

89

Cravegna & Kushmerick: ECML-2003 Tutorial

Summary and Conclusions

The summary of the summary
Where do we go from now?

89

Cravegna & Kushmerick: ECML-2003 Tutorial

Summary

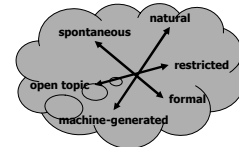
- **Information extraction:**
 - core enable technology for variety of next-generation information services
 - Data integration agents
 - Semantic Web
 - Knowledge Management
- Scalable IE systems must be **adaptive**
 - automatically learn extraction rules from examples
- **Dozens of algorithms** to choose from
- State of the art is **70-100% extraction accuracy** (after hand-tuning!) across numerous domains.
 - Is this good enough? Depends your application.
- **Yeah, but does it really work?!**
 - Several companies sell IE products.
 - SW ontology editors start including IE

31

Cravegna & Kushmerick: ECML-2003 Tutorial

Open issues, Future directions

- Knob-tuning will continue to deliver substantial incremental performance increments
- Grand Unified Theory of text "structuredness", to automatically select optimal IE algorithm for a given task

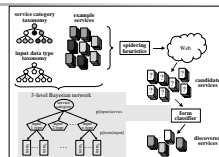


32

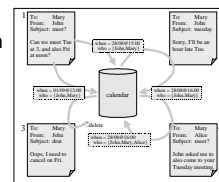
Cravegna & Kushmerick: ECML-2003 Tutorial

Open Issues, Future directions

- Resource Discovery



- Cross-Document Extraction



33

Cravegna & Kushmerick: ECML-2003 Tutorial

Open issues, Future directions

- Adaptive only?
 - Mentioned systems are designed for non experts
 - E.g. do not require users to revise or contribute rules.
 - Is this a limitation? What about experts or even the whole spectrum of skills?
 - Future direction: making the best use of user's knowledge
- Expressive enough?
 - What about filling templates?
 - Coreferences (ACME is producing part for YMB Inc. The company will deliver...)
 - Reasoning (if X retires then X leaves his/her company)

34

Cravegna & Kushmerick: ECML-2003 Tutorial