EUROLAN 2011 Summer School
“Babeș-Bolyai” University of Cluj-Napoca

Proceedings of the Workshop “Language Resources and Tools with Industrial Applications”

Cluj-Napoca, August 30–31, 2011

Publishing House of the Alexandru Ioan Cuza University of Iași

Volume printed with the support of the **Ministry of Education, Research, Youth and Sports** through the **National Authority for Scientific Research**
(Project number 125 M/29.07.2011)

ISSN 1843-911X

EUROLAN 2011 Summer School
"Babeș-Bolyai" University of Cluj-Napoca

Proceedings of the Workshop "Language Resources and Tools with Industrial Applications"

Cluj-Napoca, August 30–31 2011

Editors:
Adrian Iftene
Diana-Maria Trandabăț

Publishing House of the Alexandru Ioan Cuza University of Iași

PROGRAM COMITEE

Dan Cristea, “Al. I. Cuza” University of Iași, Faculty of Computer Science and Institute for Computer Science, R.A., Iași

Maria Husarciuc, “Monumenta linguae Dacoromanorum”, Center for Bilical Philological Studies “Al. I. Cuza” University, Iași

Adrian Iftene, “Al. I. Cuza” University of Iași, Faculty of Computer Science

Alex-Mihai Moruz, “Al. I. Cuza” University of Iași, Faculty of Computer Science and Institute for Computer Science, R. A., Iași

Ionuț Pistol, “Al. I. Cuza” University of Iași, Faculty of Computer Science

Diana Maria Trandabăț, “Al. I. Cuza” University of Iași, Faculty of Computer Science and Institute for Computer Science, R. A., Iași

FOREWORD

The new information and communication technologies have changed the business management climate. Organizations use everyday technological innovations to become more competitive. With the rapid development of natural language processing techniques, as well as their use in ever more industrial applications, the link between companies and academia is more necessary than ever.

In this context, the Workshop “Language Resources and Tools with Industrial Applications” provided an opportunity for graduate students (mainly PhD or MSc) and for young researchers working in the field of NLP to present their research and to indicate possible applications of their research in industrial applications.

The Workshop “Language Resources and Tools in Industrial Applications” was held jointly with the EUROLAN 2011 Summer School – Natural Language Processing Goes Industrial. The EUROLAN 2011 summer school provided one week of intensive study of the natural language processing technologies currently under development to support industrial applications. Internationally known scholars, researchers (with the particular involvement of scientists from the Multilingual Europe Technology Alliance – META), but also industrials involved in leading-edge work in innovative areas of natural language processing gave lectures at the school (tutorials, hands-on labs and demos) to share with students in-depth understanding and experience.

The main goal of this workshop was to identify connections between research and industry, by imagining industrial scenarios in which Natural Language applications are needed. Another important gain of this workshop was the fact that the workshop audience (consisting also in lecturers that gave invited tutorials at the EUROLAN summer school) offered the presenters constructive feedback and guidance for future research.

The seven presentations were held in two days: on the 30th of August, five presentations took place, and on the 31st of August during the Sentimatrix project meeting, the last two presentations were discussed. We thank the presenters for their interesting talks and demos and to all participants for their comments and remarks during the workshop. We also want to thank the reviewers of the papers. With their effort and valuable comments, the quality of the received papers was successfully improved, and the outcome is published in this volume.

September 2011

The editors

CONTENTS

FOREWORD **5**

BOOK'S PAPERS **9**

A Contrastive Study of Syntactic Constituents in English and Romanian Texts 11
Mihaela Colhon

Hybrid POS Tagger 21
Radu Simionescu

Multilingual Mechanisms in Computational Derivational Morphology 29
Mircea Petic, Veronica Gîsca, Olga Palade

Using Natural Language Technology to Measure Mass Media Reactions in the Election Context.. 39
Daniela Gîfu, Dan Cristea

Using Textual Entailment in Internet Surveillance 49
Adrian Iftene

Towards Representation of the Discourse Structure Leading to Consensus 61
Tatiana Zidrasco, Victoria Bobicev

Using Semantic Roles to Understand the Web 69
Diana Trandabăț

INDEX OF AUTHORS **83**

PROGRAMME **85**

BOOK'S PAPERS

A Contrastive Study of Syntactic Constituents in English and Romanian Texts

Mihaela Colhon

University of Craiova, Department of Computer Science,
Al. I. Cuza Street, 13, 200585 Craiova, Romania,
mcolhon@inf.ucv.ro

Abstract. This paper addresses a contrastive study of the grammatical correspondences which can be identified between English and Romanian languages. This study is built upon an English-Romanian parallel Treebank that was previously generated by the author based on a parallel word-aligned English-Romanian corpus.

Keywords: parallel corpora, contrastive study, syntactic constituents.

1 Introduction

Machine Translation (MT) represents the usage of computers and tools for translating texts from a source language to a target language. Huge and growing demand exists for automatic translations of scientific and technical documents, commercial and business transactions, administrative memoranda, legal documentation, instruction manuals, agricultural and medical text books, industrial patents, publicity leaflets, newspaper reports, etc. [6]. In this context any resource that could help or improve automatic translations quality is important. The bilingual contrastive study of the present paper is supported by a parallel Treebank resource previously constructed upon a word-aligned bilingual corpus.

While monolingual Treebanks are widely available thanks to large-scale annotation projects (Penn Treebank [15], Prague Dependency Treebank [16], NEGRA Treebank [14]), bilingual parallel corpora with syntactic tree-based annotation on both sides, so-called parallel Treebank, are quite rare. Still, parallel Treebanks remain important linguistic resources used in a variety of NLP tasks, as it was discussed at several conferences such as *2006 International Symposium on Parallel Treebanks* and *2008 Conference of Using Corpora in Contrastive and Translation Studies*. Several systems which operate in the NLP scientific field, from Part of Speech tagging (i.e. morphological analysis) and parsing (i.e. syntactic analysis), to Machine Translation, Question Answering or Information Extraction, are trained and tested on Treebanks [2]. Generation of such Treebanks usually implies huge efforts. Manual construction is an expensive, time-consuming and error-prone process which requires linguistic expertise in both languages in question. For this reason, there has been a lot of research on automatic generation, basically using tree-to-string MT models, (e.g. [20]), while the development of tree-to-tree based MT models, despite their potential, has suffered.

Parallel Treebanks are useful not only for syntax-based MT or example-based MT but also can be exploited in statistical paradigms of translation. More precisely, by providing alignments between the syntax trees of two corresponding sentences on a sub-sentential level (word, phrase and/or clause level) automatic derivation of syntactic transfer rules, very important in any translation study, can be obtained.

Contrastive studies search for patterns of word combination, count word frequencies, register all the uses of a particular word or expression, analyze the results and elaborate theories as well as reference works on that empirical data [5]. Very helpful for such studies as parallel texts annotated accordingly to the intended study approach. The comparative study presented in this paper is based on a parallel Treebank resource and focuses on English and Romanian languages specific syntactic structure. By means of this study we intend to extract cross-linguistic structural transfer rules for English and Romanian parallel texts.

1.1 Phrasal Alignments in Parallel Treebanks

The mechanism for parallel Treebank constituents alignments, that is for syntactic tree alignments stands on phrasal alignment which can be regarded as an additional layer of information on top of the syntactic structure [19]. It shows which part of a sentence in one language is equivalent to which part of a corresponding sentence in another language and basically is a subtree alignment mechanism with only one remark: phrases shall be aligned only if the tokens (that is, the words) they span represent the same meaning. In this manner, the alignments could provide translation units outside the current sentence context [19].

In order to mention only the most recent works in this field, we remind here the Lavie's clever mathematical trick based on prime factorization to induce sub-tree alignments [11] that was superseded by Ambati and Lavie's statistical word alignment method between words in the tree pairs used to guide all hierarchical alignments which are consistent with word alignments [1].

Starting from the Lavie's works, in [4] we present an English-Romanian Parallel Treebank (shortly, ERPT) generation algorithm built upon word level alignments and parse trees for the English part of the corpus. English sentences are processed with a full syntactic parser - Stanford Parser [17]. Given this information, the algorithm starts by traversing each syntactic English tree from the leaves to the root node. For every non-terminal node *n* of an English tree, the smallest contiguous sub-sentential segment of Romanian words that are aligned with all the words in the yield (span) of the English node is determined. If such a consistent alignment is found, we project the English node on a new node in the Romanian part. The new Romanian node will be considered aligned with the node *n* and ancestor for all Romanian nodes found aligned with the English node descendants.

As we have already mentioned, ERPT generation mechanism is guided by the word alignment existing between the parallel sentences. For this reason, the mechanism is strongly dependent on the quality and quantity of the word-alignments. Referring to the Blinker annotation guidelines [13], if a word is left unaligned on the source side of a sentence pair, this implies that the meaning it carries was not realized anywhere in the target string. From the MT usage point of view, it implies that the meaning together with all the morpho-syntactic information of the source word is lost. Therefore, the more accurate the word alignments the better the quality of the induced syntax tree for the Romanian part of the corpus (Fig. 1 and Fig. 2).

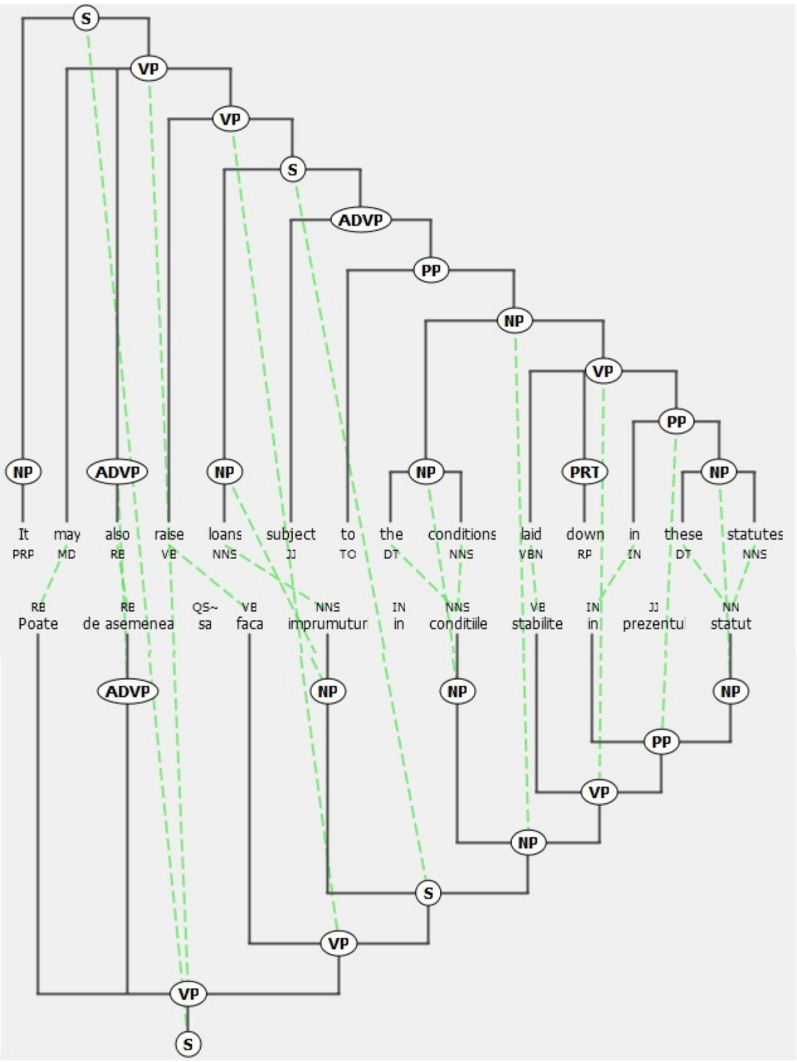


Fig. 1. ERPT Phrasal Alignments.

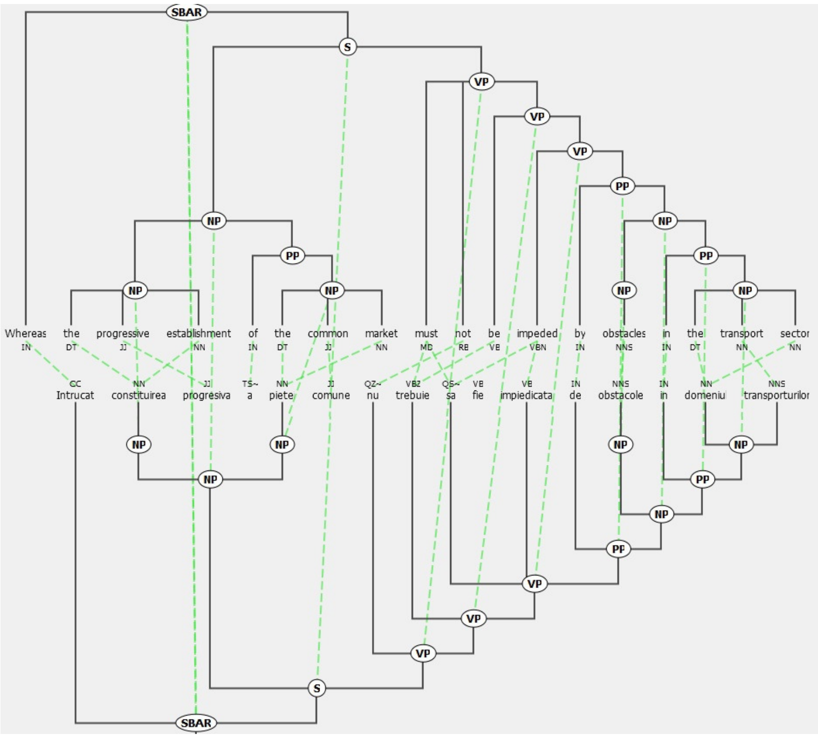


Fig. 2. ERPT phrasal alignments corresponding to a Noun Phrase (NP) prefixed and postfixed by attributes and corresponding to a negative Verb Phrase (VP) form.

2 Part of Speech Tagging for English-Romanian Texts

The morpho-syntactic annotation, also called Part-of-Speech (POS) tagging, describes the annotated words in terms of grammatical tagging (Noun, Verb, Pronoun, ...) and morphological information (e.g. gender or number). Often, POS tagging can include lemmatization, by indicating the lexemes of the words.

Most syntactic parsers use the PENN Treebank tagset [15]. The MULTEXT-East project, developed for most European languages (including Romanian), defines tagsets not only for POS information, but also for morphosyntactic descriptions - MSD [12].

The Acquis Communautaire is the total body of European Union (EU) law being available in 22 official languages (including Romanian) and represents the biggest parallel corpus existent at this moment, taking into account both its size and the number of covered languages [7]. A significant part of these parallel texts has been compiled by the Language Technology group of the European Commission into an aligned parallel corpus, called JRC-Acquis [9]. The Southeast European ERA-Net SEE-ERA.net corpus is a 1.5 million words subcorpus of JRC-Acquis corpus. The tagsets used to annotate the words of this corpus comes from the MULTEXT-East specifications Version 3 (the morpho-syntactic specifications can be found at [12]).

In order to generate ERPT, we worked with the word-aligned bilingual English-Romanian SEE-ERA.net corpus [18]. The English sentences are extracted from the corpus and processed with Stanford Parser. As a consequence, the English texts of the corpus are annotated with Penn Treebank POS and Phrasal tags as this is the tagging standard used by Stanford Parser.

We try to keep the POS tag set used for Romanian texts annotations as small and simple as possible, considering the morpho-syntactic richness of Romanian language. For this reason and also for ensuring an uniform tagging system, we reduced when it was possible, the Multext-East specifications for the Romanian texts of SEE-ERA.net corpus to their corresponding Penn Treebank tags.

In this manner, we ensure that the resulted parallel Treebank is made of isomorphic parse tree structures that could further provide accurate syntactic analysis for every translation model.

In Table 1 we present the POS tagset used in ERPT. Tags that correspond to Multext-East annotations are used in a simplified form where the character "_" replace one letter code for the corresponding tag value while the character "~" denotes terminal set of codes. For the Romanian POS tags we have taken into account the tagset proposed in [3].

Table 1. POS tags of ERPT.

<table>
<tr><th>No.</th><th>POS tag</th><th>Value</th><th>EN</th><th>RO</th></tr>
<tr><td>1.</td><td>Noun</td><td>common singular</td><td colspan="2">NN</td></tr>
<tr><td></td><td></td><td>common plural</td><td colspan="2">NNS</td></tr>
<tr><td></td><td></td><td>proper singular</td><td colspan="2">NNP</td></tr>
<tr><td></td><td></td><td>proper plural</td><td colspan="2">NNPS</td></tr>
<tr><td>2.</td><td>Verb</td><td>modal</td><td colspan="2">MD</td></tr>
<tr><td></td><td></td><td>aux</td><td></td><td>Va~</td></tr>
<tr><td></td><td></td><td>copula</td><td></td><td>Vc~</td></tr>
<tr><td></td><td></td><td>base</td><td>VB</td><td></td></tr>
<tr><td></td><td></td><td>copula</td><td></td><td>V_i~</td></tr>
<tr><td></td><td></td><td>present participle</td><td colspan="2">VBG</td></tr>
<tr><td></td><td></td><td>past participle</td><td colspan="2">VBN</td></tr>
<tr><td></td><td></td><td>gerund</td><td colspan="2">VBG</td></tr>
<tr><td></td><td></td><td>non 3rd pers sg. present</td><td colspan="2">VBP</td></tr>
<tr><td></td><td></td><td>3rd pers sg. present</td><td colspan="2">VBZ</td></tr>
<tr><td></td><td></td><td>past</td><td colspan="2">VBD</td></tr>
<tr><td></td><td></td><td>others</td><td colspan="2">VB</td></tr>
<tr><td>3.</td><td>Adjective</td><td>positive</td><td colspan="2">JJ</td></tr>
<tr><td></td><td></td><td>comparative</td><td colspan="2">JJR</td></tr>
<tr><td></td><td></td><td>superlative</td><td colspan="2">JJS</td></tr>
<tr><td>4.</td><td>Pronoun</td><td>possessive</td><td colspan="2">PRP$</td></tr>
</table>

<table>
<tr><th>No.</th><th>POS tag</th><th>Value</th><th>EN</th><th>RO</th></tr>
<tr><td></td><td></td><td>reflexive</td><td></td><td>Px~</td></tr>
<tr><td></td><td></td><td>demonstrative</td><td></td><td>Pd~</td></tr>
<tr><td></td><td></td><td>other</td><td colspan="2">PRP</td></tr>
<tr><td>5.</td><td>Determiner</td><td>general</td><td colspan="2">DT</td></tr>
<tr><td></td><td></td><td>predeterminer</td><td>PDT</td><td></td></tr>
<tr><td>6.</td><td>Article</td><td>definite</td><td rowspan="4">DT</td><td>Tf~</td></tr>
<tr><td></td><td></td><td>indefinite</td><td>Ti~</td></tr>
<tr><td></td><td></td><td>possessive</td><td>Ts~</td></tr>
<tr><td></td><td></td><td>demonstrative</td><td>Td~</td></tr>
<tr><td>7.</td><td>Adverb</td><td>positive</td><td colspan="2">RB</td></tr>
<tr><td></td><td></td><td>comparative</td><td colspan="2">RBR</td></tr>
<tr><td></td><td></td><td>superlative</td><td colspan="2">RBS</td></tr>
<tr><td>8.</td><td>Apposition</td><td>preposition <i>to</i></td><td>TO</td><td></td></tr>
<tr><td></td><td></td><td>other</td><td colspan="2">IN</td></tr>
<tr><td>9.</td><td>Conjunction</td><td>coordinating</td><td colspan="2">CC</td></tr>
<tr><td>10.</td><td>Numeral</td><td>cardinal</td><td colspan="2">CD</td></tr>
<tr><td>11.</td><td>Particle</td><td>negative</td><td rowspan="3">RP</td><td>Qz~</td></tr>
<tr><td></td><td></td><td>infinitive</td><td>Qn~</td></tr>
<tr><td></td><td></td><td>subjunctive</td><td>Qs~</td></tr>
<tr><td></td><td></td><td>other</td><td colspan="2">RP</td></tr>
</table>

3 Contrastive Study of English-Romanian Syntactic Constituents

English is a synthetic language with fewer inflexions, such as the *-s* ending for the nouns plural and for the third person of verbs and the *-ed* suffix for the Past Tense and of Past Participle for the regular verbs. Therefore, the word positions inside a sentence play an important role in the process of identify their syntactic functions.

As opposite, Romanian language has more inflexions which implies that the words position inside a sentence is not so important. Because of the inflections, certain categories of words (nouns, articles, adjectives, pronouns, numerals or verbs) can express themselves grammatical relations. As a consequence, the subject is not always indicate because the Romanian verb is formally marked for person and number (e.g. ***He*** *is not there*. versus (***El***) *nu este acolo*.)

English has two cases – the nominative (for the subject and object) and the genitive (for possession). Romanian's case system is more elaborated. There are five cases in Romanian: nominative (for subject), accusative (for direct object), dative (for the indirect object) and genitive (case of possession) and vocative. Because our study is mainly based on the Penn Treebank annotations which includes few morphological analyzes, the study of cases, moods and tenses will not be debated in this article. In sequel, we suppose that nouns are in nominative case and verbs are at present indicative.

3.1 English Syntactic Patterns vs. Romanian Patterns

Although there aren't general rules regarding how a syntactic structure in one language is transformed during translation into another language, we have identified some basic patterns by performing a contrastive English-Romanian syntactic study on the generated English-Romanian parallel Treebank.

Subject. Usually, in any declarative English sentence, subject takes the first position. Thus, the general structure of an English sentence is:

```
Subject + Predicate + Adverbs
```

Of course, there are exceptions for this rule. In imperative sentences, the subject comes after the predicate, but this kind of sentences does not make the subject of our study.

Exceptions can be found also in declarative sentences where partial inversions between subjects with the auxiliary verb that enters in the predicate construction are encountered. Inversions appear when the logical subject is preceded by an introductory subject (such as *it* or *there*).

As opposite, in Romanian the subject could take the second position after the predicate but also, it can be omitted at all.

Attribute (or adjectival part of speech). Usually, adjective precedes the part of speech that it modifies. Playing the role of attribute we find adjectives, nouns, demonstratives, numerals and participles. For the English language there are some exception regarding the precedence of attribute. There are several cases in which the subject or other nominal element of a syntactic phrase is followed by its attribute:

- the adjectives like *present*, *proper* and *extant*: (e.g. *the people present in the room*);
- in fixed multi-word constructions, like *sum total*;
- the adjectives used in a predicative form like *the house ablaze*;
- with indefinite pronouns ending in *-body*, *-one*, *-thing* like *nothing unusual*;
- in prepositional constructions (noun + preposition) like *picture of my son*;
- in infinitive constructions like *the question to be settled*.

For the Romanian language, the descriptive adjective is usually placed after the noun it modifies and takes the gender, number and case of the modifier noun(s). There are few exceptions of this rule: nouns in vocative are preceded by adjectives as in *dragă prietene* (En: dear friend) and all emphatic constructions, where the whole structure "adjective + noun" indicates a strong emotional involvement of the speaker/narrator.

Predicate. In the English language, for predicates specified by verbs at present indicative, the general rule is that predicate takes the second place in a sentence, after the subject.

The exceptions refer to the case when the predicate is preceded by some indefinite frequency adverbs (e.g. *often*, *always*, *usually*, *never*, *rarely*, *seldom*, *sometimes*) like in *they rarely visit us* or restrictive adverbs (*hardly*, *scarcely*, *only*).

The transitive verbs are followed by direct objects while the intransitive verbs are followed by adverbs or adverbial phrases of manner, place and time (in this exact order).

Romanian verbs have different forms that show mood, tense, person, number, gender and voice. The negative of all the Romanian verbs is formed with the negation *nu* (En: not) placed before the verbal form, simple or compound.

Adverbs are invariable parts of speech that usually accompany and modify verbs (they also can modify adjectives or other adverbs). Romanian language places the adverb before the modified verb. When used with verbal forms without auxiliaries, they just precede the verb. When combined with negative forms, the adverbs come after the negation *nu*. When used with verbal forms with auxiliaries, some adverbs (like *tot*, *mai*, *cam*, *prea* and *și* (En: all, more, some, very and and)) are placed between the auxiliary and the verb (e.g. *Am mai fost pe aici* (En: I have been here.)).

Adverbs can be accompanied by prepositions *de*, *până*, *pe*, *pentru*, etc. (En: of, up, on, for). Romanian adverbial phrases are numerous, but they lack an adverb in their structure. Their meaning and function are established taking into account the whole construction, and for this reason these phrases are often considered as adverbial collocations (examples of Romanian adverbial phrases: *fără îndoială*, *cu siguranţă* (En: no doubt, certainly)). Units that turn out to be collocations will not be studied for their syntactic structure because this study is not relevant in this case.

4 Future Work

Creating a parallel Treebank is a laborious and time-consuming task. Each step in the creation process needs to be inspected before the next step is carried out. The main function of such resource is to support research for cross-linguistic syntactical phrases transfer rules.

However, the syntactic phrases on one language text do not necessarily map to syntactic phrases on the parallel text. Still, we intend to extract as many structural rules as possible to address coverage, using statistical translation techniques. The final goal is to show syntactic phrase translation equivalence.

Acknowledgments. The author M. Colhon has been funded for this research by the strategic grant POSDRU/89/1.5/S/61968, Project ID 61986 (2009), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007-2013.

References

1. Ambati, V., Lavie, A.: Improving Syntax Driven Translation Models by Re-structuring Divergent and Non-isomorphic Parse Tree Structures, In: AMTA (2008)
2. Bosco, C.: A Grammatical Relation System For Treebank Annotation, Ph.D. Thesis, Dipartimento di Informatica Universita degli Studi di Torino, Italy (2001)
3. Călăcean, M.: Data-driven Dependency Parsing for Romanian. In: Master's thesis in Computational Linguistics, Uppsala University (2008)
4. Colhon, M.: Parallel Treebank From Word-Aligned Bilingual Corpus. Language Engineering for Phrasal Alignments. In. 15th International Conference on System Theory, Control and Computing ICSTCC 20011, Sinaia, Romania (2011)
5. Göhring, A.: Spanish Expansion of a Parallel Treebank. In: Ph.D. Thesis, University of Zürich, Switzerland (2009)
6. Hutchins, W. J., Somers, H. L.: An Introduction to Machine Translation. In: Academic Press (1992)
7. Irimia, E: EBMT Experiments for the English-Romanian Language Pair. In: Recent Advances in Intelligent Information Systems, 91-102 (2009)
8. Johansson, S.: Contrastive Linguistics and Corpora. In: SPRIKreports (3), Departament of British and American Studies, University of Oslo (2000)
9. JRC-Acquis, http://langtech.jrc.it/JRC-Acquis.html
10. Kromann, M. B., Hardt, D., Korzen, I.: Syntax-Centered and Semantics-Centered Views of Discourse. Can They be Reconciled? In: Workshop Beyond Semantics: Corpus-based Investigations of Pragmatic and Discourse Phenomena, Bochumer Linguistische Arbeitsberichte (3), Germany, 17-30 (2011)
11. Lavie, A., Parlikar, A., Ambati, V.: Syntax-driven Learning of Subsentential Translation Equivalents and Translation Rules from Parsed Parallel Corpora. In: SSST-2, Columbus, Ohio, 87-95 (2008)
12. MULTEXT-East specifications, http://nl.ijs.si/ME/V3/msd
13. Melamed, I.D.: Manual Annotation of Translational Equivalence: The Blinker Project. In: IRCS Technical Reports Series, University of Pennsylvania, (1998)
14. NEGRA Treebank, http://www.coli.uni-sb.de/sfb378/negra-corpus/
15. Penn Treebank, http://www.cis.upenn.edu/ treebank/home.html
16. Prague Dependency Treebank, http://shadow.ms.mff.cuni.cz/pdt/
17. Stanford Parser, http://nlp.stanford.edu/software/lex-parser.shtml
18. Tufiş, D., Koeva, S., Erjavec, T., Gavrilidou, M., Krstev, C.: Building Language Resources and Translation Models for Machine Translation Focused on South Slavic and Balkan Languages. In: J. Machacova and K. Rahsmann (eds) "Scientific Results of the SEE-ERA.NET Pilot Joint Call", Center for Social Innovation Publisher, Vienna, 37-48 (2009)
19. Samuelsson, Y., Volk, M.: Alignment Tools for Parallel Treebanks. In: Proc. of the Linguistic Annotation Workshop (LAW) at ACL (2007)
20. Yamada, K., Knight, K.: A Syntax-based Statistical Translation Model. In: Proc. of the Conference of the ACL (2001)

Hybrid POS Tagger

Radu Simionescu

"Al. I. Cuza" University of Iasi, Faculty of Computer Science
radu.simionescu@info.uaic.ro

Abstract. This work presents a hybrid part-of-speech (POS) tagger which successfully combines a statistic model with a rule based system. Common POS taggers usually have a POS dictionary for reducing tagging ambiguity before classifying the input words. The rules system described is used to reduce this ambiguity even further. Linguist experts can define rules to help increase the tagging precision. We provide a rule editing tool which detects frequent error types while evaluating the tagger. This tool can also detect frequent fail cases generated only by a certain rule alone. Finally, we present the results of the model for a Romanian Hybrid POS tagger.

Keywords: POS, tagging, part of speech, grammar, hybrid, constraints.

1 Introduction

When a simple statistic part of speech (POS) tagger generates a type of error systematically, the only solution to fix it is to tweak various parameters. This usually leads to a trial and error approach which is not guaranteed to fix the problem. Modifying some of the parameters might also require retraining of the entire statistic model, which can take a lot of time.

By detecting the linguistic conditions for which the classifier generates a type of error, the hybrid model described in this paper can be configured to fix these errors with little effort.

Most common POS taggers use a POS dictionary for constraining each word to a small set of possible output tags. Then, a statistic model is used to disambiguate among these. The Hybrid POS tagger presented uses a method of reducing the ambiguity even further, by using rules which can take into account any features of the words within the input sentence.

POS taggers sometimes fail to correctly classify cases for which linguists can easily decide the correct part of speech. These types of errors are generated due to noise in the training data but also because a machine learning method cannot yet detect all the linguistic phenomena in the training set. The main goal of this work is to overcome this limitation.

2 The Rules System

Before applying the rules, each word of the input sentence is associated with a list of possible tags, based on the POS dictionary. If a word is not found in the POS dictionary, it is associated with a predefined list of tags, considered the guesser tagset.

Applying a rule on a word, results in the reduction of the number of the possible POS tags for some tokens (usually, for the token for which it is applied, but other tokens can also be affected). All rules are applied on all words, from left to right.

To implement the rules system, we created a language for describing the rules. Each rule is composed of two parts: condition and actions. The main idea is similar to constraint grammars (CG) [4] and JAPE [2] but this grammar description language was created for writing tagging disambiguation rules specifically, while CG and JAPE are for wider use.

The conditions are specified as a sequence of words which have certain features (internal conditions). Describing a condition requires fundamental knowledge of regular expressions. The actions are specified as a set of "KEEP" or "REMOVE" commands.

2.1 Specifying Conditions

The following writing conventions are used:

- **<<>>** – current token
- **<>** – a normal token

A condition is a sequence of tokens, and one of them must be identified as the current one.

- **<...>{3,7}** – a token which matches 3, 4, 5, 6 or 7 times;
- **<@name (...) (...) ...>** – associates the token with the **@name** identifier; **@name** can be used later in the actions section;
- **<(()())>** , **<(()|())>** – conjunction and disjunction of internal conditions.

Internal conditions for matching tokens are presented below:

- **<... (POS === /expr/) ...>** – tests if the list of possible tags contains at least one tag which matches the regular expression *expr*;
- **<... (POS$var === /expr/) ...>** – if the test succeeds, the variable **$var** will contain the tags which matched *expr*; **$var** can be used later in the actions section;
- **<... (!POS === /expr/) ...>** – negation – tests if the list of possible tags doesn't contain any tag which matches *expr*;
- **(WORD === /expr/)** – tests if *expr* matches with the literal form of the word;

- **(INDIC)** – test if the word has been found in the dictionary.

To match the sequence of tags, when compiling a rule, two state machines are created: one for the sequence to the left of the current token, and one for the sequence to the right.

- **!<<>>** and **<<>>!** – (an exclamation mark to the left or right of the current token) negates the test result for the sequence to the left or right of the current token (necessary for testing if a sequence doesn't exist; it is similar with the "negative assertion" operation, from regular expressions).

2.2 Specifying Actions

- **->** – marks the end of the condition sequence section and the beginning of the actions;
- **<KEEP @tok $var_k ... literal_k ...>** – keep only **$var_k, literal_k, ...** in the list of possible tags for token **@tok**;
- **<REMOVE @tok $var_k ... literal_k ...>** – remove **$var_k literal_k ...** from **@tok.**

Examples

1. <(WORD===/the|a/i)> <<@c (POS === /NN/)>> -> <KEEP @c NN>;
   <(WORD===/the/i)> <<@c (POS === /NNs/)>> -> <KEEP @c NNs>.

If the current token is preceded by an article, and it can be classified as a noun, than it can only be a noun.

2. <(WORD===/the|a/i)> <(POS===/JJ.*/)><0,3> <<@c (POS === /NN/)>> -> <KEEP @c NN>.

This example takes the previous rules a step further. It is applied even if there are a maximum of 3 adjectives between the article and the current token (e.g. "the small tender little fly").

3. <><<@c((!INDICT)(POS $npPOS === /Np.*/)(WORD===/^\p{Lu}.*/))>> -> <KEEP @c $npPOS>.

If the current token was not found in the dictionary and it starts with an upper case letter and it can be tagged as "Np.*" (proper noun) and it is not the first in the sentence, then the current token can only be a proper noun.

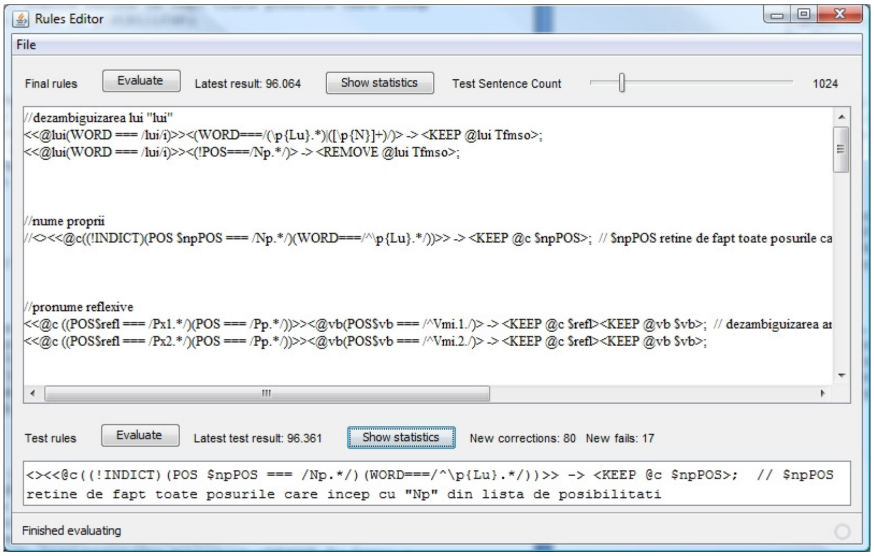
The following example is relevant for Romanian

4. <(WORD === /(să)/i) > <>{0,5} !<<@c(POS$conj===/^Vms.*/)>> -> <REMOVE @c $conj>.

If the current word can be classified with a tag which starts with "Vms" (which stands for conjunctive verb) and if there is no word "să" (which is a mark for the conjunctive verb in Romanian) to the left at a distance of maximum 5 words, then remove the tags which start with “Vms” from the list of possible tags of the current token. In other words, the current token cannot be a conjunctive verb if there is no word “să” to its left, at a distance of maximum 5 tokens. This rule is just an example. In practice, when writing such a rule, we would also constrain it so that it wouldn’t apply if there was any other possible conjunctive verb between „să” and the current token.

3 A Rule Editing Tool

Writing rules to enhance the precision of the POS tagger can be difficult without any information about the errors generated at evaluation. For this reason we developed an application which, while evaluating the model on a test corpus, shows various statistics regarding the fail cases detected. The software also provides the functionality to evaluate only a certain rule (Test rule) and see the increase or decrease of precision determined by it (Fig. 1.). This tool is completely language independent.



Latest result: 96.064

Test Sentence Count 1024

```
//dezambiguizarea lui "lui"
<<@lui(WORD === /lui/i)>><(WORD===/(\p{Lu}.*)([\p{N}]+)/)> -> <KEEP @lui Tfmso>;
<<@lui(WORD === /lui/i)>><(!POS===/Np.*/)> -> <REMOVE @lui Tfmso>;



//nume proprii
//<><<@c((!INDICT)(POS $npPOS === /Np.*/)(WORD===/^\p{Lu}.*/))>> -> <KEEP @c $npPOS>;  // $npPOS retine de fapt toate posurile ca



//pronume reflexive
<<@c ((POS$refl === /Px1.*/)(POS === /Pp.*/))>><@vb(POS$vb === /^Vmi.1./)> -> <KEEP @c $refl><KEEP @vb $vb>;  // dezambiguizarea ar
<<@c ((POS$refl === /Px2.*/)(POS === /Pp.*/))>><@vb(POS$vb === /^Vmi.2./)> -> <KEEP @c $refl><KEEP @vb $vb>;
```

Latest test result: 96.361

New corrections: 80 New fails: 17

```
<><<@c((!INDICT)(POS $npPOS === /Np.*/)(WORD===/^\p{Lu}.*/))>> -> <KEEP @c $npPOS>;  // $npPOS
retine de fapt toate posurile care incep cu "Np" din lista de posibilitati
```

Finished evaluating

Fig. 1. Rules Editor.

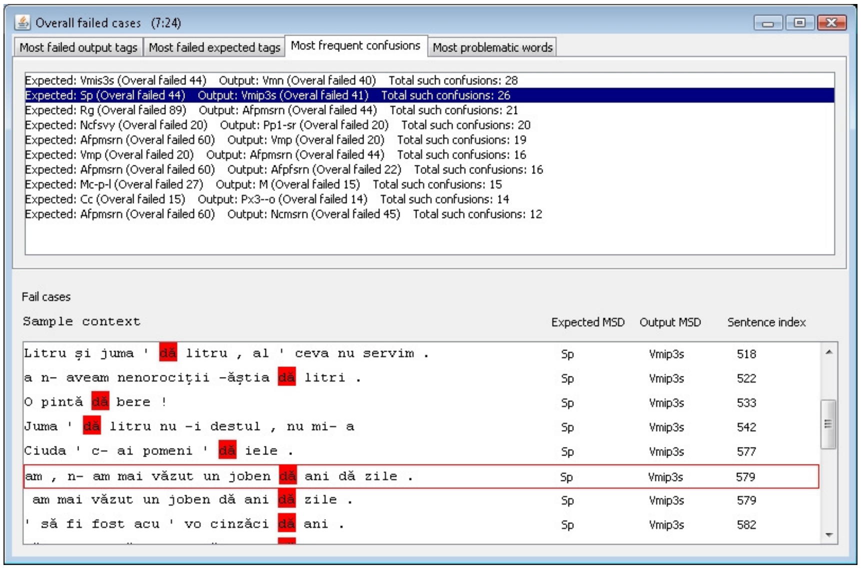
The evaluation of the model when using only the “Final rules” is a reference point for determining the differences created later by the “Test rule”. Evaluating the Test rule results in evaluating using the “Final rules” plus the “Test rule”. The statistics for the “Test rule” are made using only the fail cases which were tagged differently. The interface classifies the fail cases by the following criteria:

- Most failed output tags;
- Most failed input tags;

  - Most frequent confusions;
  - Most problematic words.

For each class of errors the user can visualize the sequences of words from the test corpus for which the POS tagger has failed (see Fig. 2.).

By analyzing the overall fail cases, a linguist can detect contextual features for solving frequent errors, and write a new rule. This can then be easily refined because the user can see the exact fail cases and statistics for the new rule. This tool is very useful for detecting various frequent morphological ambiguities. By observing the fail cases in the test corpus, one can often find contextual features to solve these.



```
Expected: Vmis3s (Overal failed 44)    Output: Vmn (Overal failed 40)    Total such confusions: 28
Expected: Sp (Overal failed 44)    Output: Vmip3s (Overal failed 41)    Total such confusions: 26
Expected: Rg (Overal failed 89)    Output: Afpmsrn (Overal failed 44)    Total such confusions: 21
Expected: Ncfsvy (Overal failed 20)    Output: Pp1-sr (Overal failed 20)    Total such confusions: 20
Expected: Afpmsrn (Overal failed 60)    Output: Vmp (Overal failed 20)    Total such confusions: 19
Expected: Vmp (Overal failed 20)    Output: Afpmsrn (Overal failed 44)    Total such confusions: 16
Expected: Afpmsrn (Overal failed 60)    Output: Afpfsrn (Overal failed 22)    Total such confusions: 16
Expected: Mc-p-l (Overal failed 27)    Output: M (Overal failed 15)    Total such confusions: 15
Expected: Cc (Overal failed 15)    Output: Px3--o (Overal failed 14)    Total such confusions: 14
Expected: Afpmsrn (Overal failed 60)    Output: Ncmsrn (Overal failed 45)    Total such confusions: 12
```

Sample context	Expected MSD	Output MSD	Sentence index
Litru şi juma ' dă litru , al ' ceva nu servim .	Sp	Vmip3s	518
a n- aveam nenorociţii -ăştia dă litri .	Sp	Vmip3s	522
O pintă dă bere !	Sp	Vmip3s	533
Juma ' dă litru nu -i destul , nu mi- a	Sp	Vmip3s	542
Ciuda ' c- ai pomeni ' dă iele .	Sp	Vmip3s	577
am , n- am mai văzut un joben dă ani dă zile .	Sp	Vmip3s	579
am mai văzut un joben dă ani dă zile .	Sp	Vmip3s	579
' să fi fost acu ' vo cinzăci dă ani .	Sp	Vmip3s	582

Fig. 2. Most Frequent Confusions.

The application doesn't require multiple initializations of the POS dictionary and the statistic model, and that is why the work flow is smooth. Evaluating usually takes less than a few seconds. Nevertheless, the interface offers the possibility to use only a fraction of the testing corpus, to speed things up, if necessary.

Please note that the software is completely language independent.

4 Romanian Hybrid POS Tagger

We developed a Romanian POS tagger using the hybrid model presented. We first constructed a POS dictionary, using the database provided by DexOnline.ro[1]. Many morphological classes were corrected and entered manually, especially for functional words (prepositions, pronouns, articles etc.). The dictionary contained 1.15 millions of words. Then we inserted another 100,000 proper nouns representing persons, cities, companies and countries, extracted from Wikipedia[2].

The lists associated with each entry contain, in total, 2.3 million tags. Each possible tag for a word has a lemma associated. The dictionary contains 230,000 distinct lemmas. According to the POS dictionary, 630,000 words are unambiguous and, from a lemmatization point of view, 82.14% of the words are unambiguous.

At this step, we also established a tagset. This is based on the MSD classification [3] used in MULTEXT-East[3], and it is a reduced version of the one used by the Research Institute for Artificial Intelligence, Romanian Academy[4] [8], which contains about 600 tags. Our tagset contains 406 tags.

The training corpus used is composed of:

- NAACL 2003[5] – approx. 39,000 sentences of newspaper text; approx. 750,000 tokens tagged with the RACAI tagset;
- Approx. 28,000 sentences extracted from JRC-ACQUIS[6] and tagged with the RACAI POS-tagger.

Training data contains 1.8 million tokens, from which 39.18% are unambiguous and 10.94% are unknown, according to the POS dictionary.

The test corpus is a partially corrected version of the Multext-East “1984” corpus. This is manually tagged with the RACAI tagset, it contains approx. 6,000 sentences and 117,000 tokens from which 42.75% unambiguous words and 8.82% unknown words.

The features used by the maximum entropy model for predicting a tag for a word are show bellow (some of these were suggested in [1]):

- Word form;
- Presence of upper case letter (in the middle or at the beginning);
- Presence of digits;
- Previous two words, and the next word;
- The tags predicted for the previous two tokens;

[1] Dex Online: http://www.dexonline.ro is the digitalization of some prestigious Romanian dictionaries. Part of the database used by DexOnline is available under the GNU license

[2] Romanian Wikipedia: http://ro.wikipedia.org

[3] ME: http://nl.ijs.si/ME

[4] RACAI: http://www.racai.ro referred in the NLP community as “RACAI”

[5] A parallel corpus for Romanian-English created at the HLT/NAACL 2003 workshop, titled “Building and Using Parallel Texts: Data Driven Machine Translation and Beyond” (http://www.cse.unt.edu/~rada/wpt05/)

[6] JRC-ACQUIS is the largest parallel corpus. It is composed of lows for the EU Member States, since 1958 till present, translated and aligned for 23 languages

- The tags which will be predicted for the next two tokens, if these are unambiguous words;
- Word suffixes of length 1, 2, 3 and 4.

For tokens which are not found in the POS dictionary, the following additional features are taken into account:

- Presence of a hyphen (beginning, middle, end);
- Token length;
- The punctuation at the end of the sentence;
- The second next word.

The statistic model we used is the maximum entropy model, but the hybrid POS tagger described is not limited to any machine learning approach. In his famous work [5] Adwait Ratnaparkhi describes the use of maximum entropy for POS tagging. The maximum entropy model is used by the state of the art POS tagger for English – Stanford Tagger – having precision of 97.32% [7]. Only recently has this score been overtaken (97.50%) [6].

When tagging a sentence, the model classifies words from left to right. Each prediction requires a set of context features for each word. Some of these features are the predictions given by the model for previous tags and thus the classification of the current word depends on the outcome of the previous. For this reason, discovering the most probable sequence of tags for an input sentence cannot be done in a greedy manner ("best first search"); exhaustive search on the other hand, is too slow. We use instead an in-between approach called "beam search". This algorithm covers only the most promising *n* paths of the search tree, where *n* is called "the width of the beam". Other known approaches parse the input sentence in both directions [9] or use a cyclic dependency network [7].

An online version and a web service description of the Romanian hybrid POS tagger are available at http://solaria.mooo.com/WebPosTagger/. We are currently looking forward to publish it as an open source project.

5 Results

Table 1 describes the precisions obtained with and without applying the rules for the implemented Romanian Hybrid POS tagger.

Table 1. Evaluation of Romanian Hybrid POS Tagger.

Precision	Without rules	With rules
For unknown words	88.88%	92.34%
For all words	95.12%	**96.66%**

The difference of 1.54% means that 31% of the errors generated by the statistic model are eliminated by the rules system. There are a total of 19 rules used in this test, and some are more efficient than others. The final results demonstrate that combining a statistic model with the rules system presented, can result in an important enhancement of the tagging precision.

The possibility to control the output of a statistic model with ease, represents a great opportunity of enhancing it. This way, many types of errors, including those generated by noise or human errors, can be fixed.

References

1. Ceauşu, A.: Maximum Entropy Tiered Tagging. Proceedings of the Eleventh ESSLLI Student Session Janneke Huitink & Sophia Katrenko (2006)
2. Cunningham, H., Maynard, D., Tablan, V.: JAPE: a Java Annotation Patterns Engine (Second Edition). Technical report CS-00-10, University of Sheffield, Department of Computer Science (2000)
3. Erjavec, T.: MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In Proc. of the Fourth Intl. Conf. on Language Resources and Evaluation, LREC'2004, ELRA (2004)
4. Karlsson, F., Voutilainen, A., Heikkilä, J., Anttila, A.: Constraint Grammar: A Language-Independent System for Parsing Running Text. Natural Language Processing, No 4. Mouton de Gruyter, Berlin and New York. ISBN 3-11-014179-5 (1995)
5. Ratnaparkhi, A.: A Maximum Entropy Model for Part-Of-Speech Tagging. Philadelphia: University of Pennsylvania Dept. of Computer and Information Science (1998)
6. Søgaard, A.: Semi-supervised condensed nearest neighbour for part-of-speech tagging. Portland, Oregon: The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT) (2011)
7. Toutanova, K., Klein, D., Manning, C. D., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, 252-259 (2003)
8. Tufiş, D.: Using a Large Set of EAGLES-compliant Morpho-Syntactic Descriptors as a Tagset for Probabilistic Tagging. International Conference on Language Resources and Evaluation LREC'2000, Athens, 1105-1112 (2000)
9. Yoshimasa, T., Jun'ichi, T.: Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data. Proceedings of HLT.EMNLP 2005, 467-474 (2005)

Multilingual Mechanisms in Computational Derivational Morphology

Mircea Petic, Veronica Gîsca, Olga Palade

Institute of Mathematics and Computer Science of the A.S.M,
5, Academies str., Chisinau, MD-2028, Republic of Moldova
{petic.mircea, veronica.gisca, olga.palade}@gmail.com

Abstract. The article presents a study on common processing mechanisms that can be applied for different languages. The paper presents approaches used in derivative recognition and segmentation and also generative models. The investigation continues with several approaches in automatic new generated words validation. A special compartment deals with the algorithm of automatic lexical derivation.

Keywords: derivation, generative mechanisms, lexical resources.

1 Introduction

Derivational morphology represents an important issue in lexical resources extension. The particularities of the derivational morphology mechanisms help in lexical resources extension without any semantic information [1]. Moreover, there are processing mechanisms similar for different languages spoken in Europe, namely English, French, Spanish, Russian, Romanian.

Based on the examples of the languages stated above the article is structured as follows: first we will present the derivatives recognition and segmentation approaches then the generation mechanisms are described.

The approaches and mechanisms presented in the paper have been studied on the examples from Romanian language, but in majority of cases can be applied to different languages. The obtained results of the investigation can be used in different domains of natural language processing, namely: stemming, language detection, machine translation, information retrieval.

2 Automatic Derivative Recognition

In the process of derivative recognition it is possible to discover correct words that are not attested in the dictionaries. In addition the derivative recognition corresponds to affix detection, but an affix belongs to a concrete language. So it can be helpful this fact to language detection. That is why it is important to have a mechanism for derivatives recognition.

As a source for automatic derivatives recognition, a lexicon serves, containing not only graphic representation of the Romanian words, but also their part of speech. The lexicon consists of approximately 100,000 of words bases, and words can have several entrances for different parts of speech. Besides the lexicon, lists of prefixes with their phonological forms and suffixes were used.

Since not all the words end (begin) with the same suffixes (prefixes), some algorithms were elaborated for enabling the automatic extraction of the derivatives from the lexicon. The elaborated algorithms took into account the fact that being $x, y \in \Sigma^{+}$, where Σ^{+} is the set of all possible roots, and if $y = xv$ then v is the suffix of y and if $y = ux$ then u is the prefix of y. In this context both y and x must be valid words in Romanian language, and u and v are strings that can be affixes attested for Romanian language. The problem of consonant and/or vowel alternations was neglected in the case of the algorithm derivatives extraction. This fact does not permit the exact detecting of all derivatives [2].

Being more precise, the following word formation scheme expresses the particularities of prefixation:

$$\left[prefix[stem]_{p}\right]_{p}$$

where p represents the part of speech for the stem and the derivative. Note that in the process of prefixation there is not a part of speech changing. In the process of suffixation there are cases of part of speech changing (for example, (a) citi → citi*tor*, in Romanian, (to) read → read*er*, in English), as it is presenting in the following word formation scheme:

$$\left[[stem]_{p1}suffix\right]_{p2}$$

Taking into consideration the peculiarities of the Romanian affixes and derivatives the algorithm for automatic derivatives recognition was elaborated.

3 Solving the Problem of Derivative Segmentation

One of the main problems concerning derivation in different languages spoken in Europe is the uncertainty of morph boundaries. In many cases different people, or even the same person in different situations, divide the same word into segments in different ways. Besides the segmentation in **morphemes**, for example, *anti-rachetă* (in Romanian, missile in English), *un-reach-able* (in English), or the allomorph variants, a word form can be segmented in different ways [1] such as: **syllables**, for example, *an-ti-ra-che-tă* (in Romanian), *un-reach-a-ble* (in English), **sounds**, for example, *a-n-t-i-r-a-ch-e-t-ă* (in Romanian), *u-n-r-ea-ch-a-b-le* (in English), and **letters**, for example, *a-n-t-i-r-a-c-h-e-t-ă* (in Romanian), *u-n-r-e-a-c-h-a-b-l-e* (in English). All these types of segmentation have nothing in common with segmentation in morphemes. The segmentation in morphemes implies the detection of the morphemes with its types (root, prefix, and suffix). Unfortunately there are not all the morphemes different. There are morphemes where the root, prefix and suffix coincide, for example in Romanian: an in *an*istoric (En: historical year) is the prefix, in *an*işor (En: year old) is the root in Americ*an* is the suffix.

The problem of segmentation in morphemes was solved with the help of a lexicon. In the case of the lexicons they are not simple repositories only of words, but they need to contain the prefixes and/or suffixes with their descriptions [3].

Table 1. The Tables' Characteristics.

Characteristics	Number
derivates	15,300
roots/stems	6,800
prefixes	42
suffixes	433

To work with affixing, we take the correspondent information from *eDCD* (electronic variant of derivatives dictionary) and added four tables to the DB: prefixes, suffixes, roots-stems-derivatives, and the table which maps affixes to roots/stems in order to form the derivatives (Table 1). The last table consists of 3 fields for prefixes and 4 for suffixes, because the electronic variant of derivatives dictionary has derivatives with maximum 2 prefixes, for example, dez/ră/suci (En: untwist), pre/în/noi (En: restore), or 3 suffixes, for example, loc/al/iza/re (En: localization).

With this structure attached to Reusable Resources for the Romanian Language (RRRL), it was possible to elaborate some queries that allow: derivative extraction of a prefix or suffix; lexical family extraction for a root or stem; the part of speech establishing of the derivatives and/or roots-stems; determining the alternations that are present in the process of derivation.

This approach is useful not only for Romanian language and helps in other preprocessing that are connected with derivative words. The corresponding lexicon is to be enriched permanently to satisfy the increasing needs of developed natural language applications.

4 Derivative Generation

The studies of the lexical derivational process conclude that it is impossible to develop a universal derivational algorithm, which would allow a practical implementation of a generator of derivatives for all opportunities. If we raise the problem of obtaining a considerable coverage of derivatives lexicon, we can consider two approaches:

- *declarative* - in this case we store all derivatives, obtained a priori from certain sources including manual derivation;

- *procedural* - derivatives are obtained in a special automatic way from the roots and themes [4].

The subject of our research is the procedural method. From the above we conclude that lexicons completion can be achieved by automatic means taking into account the productive properties of derivational processes. Thus the basis for generating new derivatives is an existing lexicon. The lexicon should contain not only graphical representation of the words, but also its parts of speech.

4.1 Stages in Procedural Lexical Derivation

Let us examine the process of lexical family generation. The problem is formulated as follows: the set of all possible Cartesian products $P \times R \times S$, where P is the set of prefixes, R - set of roots, S - set of suffixes, we construct several subsets:

- products that can be generated by concatenation $[p]r[s]$, where $p \in P$, $r \in R$, $s \in S$, and [.] denotes that the particle may be missing;
- products that can be generated by concatenation $[p]r'[s]$, where $p \in P$, $s \in S$, and $r' \in R' \subseteq R$ - root set of possible alternations;
- products $[p']r[s']$, where $p' \in P' \subseteq P$, $s' \in S' \subseteq S$, concatenation will not form valid words of a natural language.

Thus, the automatic derivation process involves three important steps:

1. **Establishing if the word is susceptible for derivation** – check whether a sequence of characters represents a correct Romanian word (we used RRRL and the possibilities of Internet) and belongs to the part of speech which could be derived (for example in Romanian, the prefix *re-* and the suffix *-tor* can be attached only to verbs, the same in English for the suffix *-able*). There is established a table of correspondence between affixes and part of speech.

2. **Derivative models application** – the most important derivative models [5] are the following:

   Affixes substitution - the usage of corresponding affixes in the process of replacement.

   a. Prefix substitution - let be x_1 a word of the form $x_1 = \alpha_1\omega$, where α_1 is a prefix. After the substitution $\alpha_1 \rightarrow \alpha_2$ we obtain the word $x_2 = \alpha_2\omega$, where x_2 is the obtained derivative, for example, *în*chide - *des*chide (in Romanian, close - open in English).

   b. Suffix substitution - let be x_1 a word of the form $x_1 = \omega\beta_1$ with the suffixe β_1. After the substitution $\beta_1 \rightarrow \beta_2$ we obtain the word $x_2 = \omega\beta_2$, for example, corig*ență* - corig*ent* (in Romanian, second examination - the pupil who has to go for a second examination in English), amortizar - amortizable (in Spanish, redeem - redeemable in English).

Derivatives projection - mixtion of affixes in the case of prefixation or suffixation of the roots. So, let be ω a word, α - a prefix, β - a suffix, then the following relations are valuable:

$$(\omega \to \alpha\omega)\wedge(\omega \to \omega\beta) \Rightarrow (\omega \to \alpha\omega\beta),$$

for example in English, (read → unread) ∧ (read → readable) ⇒ (read → unreadable).

Formal derivation rules - rules that depend on graphic representation of a word and its part of speech can lead to derivatives generation of a high degree of accuracy, for example from the infinitive by adding the prefix *re-* (in Romanian, English, French), we obtain other verbs that mean the repetition of the action.

Derivational constraints - some schemes with several parameters that reduce the class roots and affixes in order to form derivatives. For example functions of the form:

$$f{:}\,\{wrd, pos, mod, sla, fgw, mvca\} \to derivative$$

where *wrd* is a word to derivate, *pos* - part of speech of *wrd*, *mod* - model of derivation, *sla* - the set of letters to which the affix is attached, *fgw* - flection group of *wrd*, *mvca* - modifications and vocalic or consonant alternations. For example,

$$f{:}\,\{a\ spinteca\ \text{(in engl. to slice)}, \text{verb}, \text{des} < verb > \\ , \ldots \text{s} \ldots, \text{V14}, \text{double consonant avoiding}\} \to de(s)spinteca$$

3. **Generated derivatives validation** - identification of the correctness of the Romanian generated words [6].

4.2 Problems of Derivatives Validation

Automatic derivation represents an over generating mechanism. That is why validation of generated words is needed. One of the methods of new word validation consists in manual verification of every new generated derivative as to correspond to semantic and morphologic rules. In the case of the proceeding is performed by a specialist in domain, the specific disadvantages of a manual work appear: considerable resources of time and the possibility to make mistakes [6].

Another method of validation consists of the verification of the derivatives in the existent electronic documents. There are different types of electronic documents. The first idea is to validate words using existent corpora that represent verified documents. The condition for being the panacea in the new word validation is a representative corpus.

On the other hand there are documents on Internet that are not verified, which makes them not credible. In order to make it more precise, the search over the Internet, using *Google.com* searching engine, should be made for the documents typed only in a specified language and assured the possibility to exclude word segmentation and the part of speech of the derivatives.

This validation tool divides the generated derivatives in three categories. The first one contains words that are not found by *Google.com* searching engine. The second consists of the derivatives that appear less than a frequency limit of n, in our case n = 1,000. Derivatives that are more frequent that limit n, are registered in the third group. This classification pretends that the words, that are listed more than frequency limit of n, are surely valid. Those, which are from the second group, can be valid but should be verified by specialists in linguistics. The derivatives, that are not present, could not be valid [6]. The idea of classification pretends to be a mixed method of validation, because needs only the manual verification for the words from the second category.

4.3 Algorithm of Automatic Lexical Derivation

The algorithm described below represents a polynomial complexity algorithm for automatic generation of lexical families and consists of a combination of implemented and verified subalgorithms that correspond to a concrete step of derivatives generation.

We use the following notations: cvt - word from which will be generated lexical family; D_{RRRL} – set of words from the lexicon $RRRL$; D_{eDCD} – set of derivatives of the word cvt existing in $eDCD$; D_{SA} – set of derivatives formed by affix substitution (procedure applied to D_{eDCD} words); D_{PD} - derivatives formed by derivative projection (process applied to the words included in $D_{SA} \cup D_{dECD}$); D_{CL} – set of derivatives formed by derivational constraints (process applied to the multitude of words $D_{PD} \cup D_{SA} \cup D_{dECD}$); D_{RD} – set of derivatives formed by formal derivation rules (process applied to the multitude of words $D_{CL} \cup D_{PD} \cup D_{SA} \cup$); $D_{gen} = D_{CL} \cup D_{PD} \cup D_{SA} \cup D_{RD}$ – set of words obtained by the automatic generation; D_{NEVAL} – set of words that are not found in Internet documents are considered invalid; D_{FILTER} – final set of words that require manual validation derivatives; D_{VAL} – final set of words that have a sufficient frequency in Internet documents to be considered valid and represents the lexical family of the word cvt. Given these notations we can write the corresponding algorithm in a conventional language:

```
Input: $cvt$
Output: $D_{VAL}$
   1. if ($\{cvt\} \cap D_{RRRL} = \emptyset$) then goto 3
              else goto 2;
   2. if ($cvt$ in Internet) then read-part-of-speach($cvt$);
                                   goto 3;}
              {else $D_{VAL}$ := $\{\}$; goto 7;}
   3. Generating sets of words
   3.1. $cvt \Rightarrow D_{eDCD}$;
   3.2. $D_{eDCD} \Rightarrow D_{SA}$;
   3.3. $D_{SA} \Rightarrow D_{PD}$;
```

```
3.4. $D_{PD} \Rightarrow D_{CL}$;
3.5. $D_{CL} \Rightarrow D_{RD}$;
4. $D_{gen} \coloneqq D_{CL} \cup D_{PD} \cup D_{SA} \cup D_{RD}$;
5. Automatic validation
5.1. $D_{NEVAL} \coloneqq nonval(D_{gen})$;
5.2. $D_{FILTER} \coloneqq semval(D_{gen})$;
5.3. $D_{VAL} \coloneqq val(D_{gen})$;
6. $D_{VAL} \coloneqq D_{VAL} + manualval(D_{SEMVAL})$;
7. Write($D_{VAL}$);
8. endalgorithm
```

In step 1, we check the presence of word cvt in $RRRL$ to ensure the existence of such a word in Romanian language and to be able to automatically extract its part of speech and other morphological categories for word cvt. If this word is not found in $RRRL$ the existence of the word cvt is checked by the Internet resources. If found, then states his part of speech by asking the user and passed to the next step. Otherwise, the algorithm goes to step 7.

Step 3 contains successive sets generation. Thus, derivatives of the word cvt are extracted first from $eDCD$. Next will be generated starting with the set D_{eDCD} other derivatives following the derivative models such as affix substitution, derivative projection, derivative constraints and formal derivational rules. As a result, we will get a set of automatic generated derivatives $D_{gen} = D_{CL} \cup D_{PD} \cup D_{SA} \cup D_{RD}$ (step 4).

In step 5, the set D_{gen} will be split automatically in three distinct sets: D_{NEVAL} (set of invalid derivatives) D_{FILTER} (set of "half" valid derivatives) and D_{VAL} (set of valid derivatives).

The implementation of the filtering step showed that 70% of the generated words are considered invalid words, 30% follow the step of manual validation. Step 6 consists of manual validation of the set words from D_{FILTER} and the addition of validated derivatives to D_{VAL}. Finally, 75-85% of filtered words are manually validated.

The extraction of the set of derivatives D_{VAL} is done in step 7.

5 Applicability of Computational Derivational Morphology

Following the information stated above, the lexicon completion can be realized with the help of automatic means. Starting with the derivation particularities the algorithm of derivation is applied to these words and the result is a set of derivatives. After applying the method of validation, we obtain correct words on the basis of language. These words are inflected by means of programs for inflection that result in a set of inflected words. This very set can complete the initial lexicon, making it actual.

Nevertheless after a cycle of bringing the lexicon up to date it is possible to apply another similar cycle (Fig. 1). So, after a finite number of cycles it is likely to finish the process of completion, in the end obtaining a "filled" (saturated) lexicon which will be complete from the point of view of derivation [6].

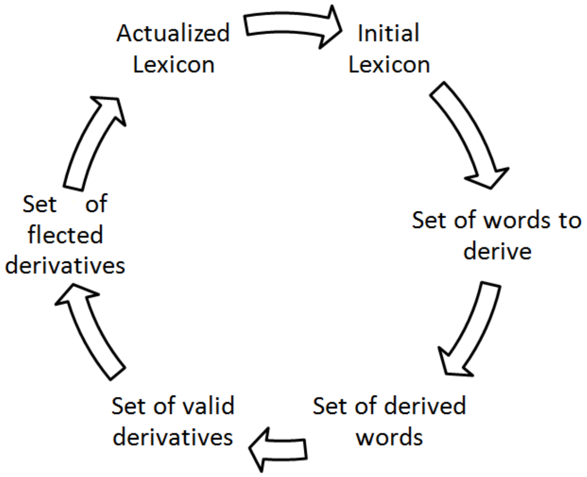


Fig. 1. Cycle of the Lexicon Completion.

The computational derivational morphology involves mechanisms solving automatic word generation. It can operatively ensure access to the fundamental linguistic resources, as well as to the resources derived by developing intelligent control mechanisms. This will simplify dictionary and lexicon completion, generation of morphological families and of words with predictable meaning. The results of this intention can be used as a support in the educational process in linguistics (or in adjacent domains such as librarianship, archive science, culturology); this will lead to reduction of costs for purchasing paper-based dictionaries and for consulting specialists. This application will allow its use both nationally (in urban areas, as well as rural areas), and abroad, which will ensure raising the level of knowledge and technologization of different languages.

6 Conclusions

The process of the derivational morphology processing needs the detailed studying of the affixes and the derivatives features. Studies on derivation process allow us to conclude that we cannot propose an effective algorithm for automatic derivation in general, but we can highlight some models of derivation, for which construction of such algorithms is possible.

The implementation in the Google engine of a module that can process derivatives will raise the accuracy of the searching queries results and will constitute a more important tool in the linguists work.

The automatic completion cycle model for lexical resources by the derivational and inflectional mechanisms allows the consciousness of the steps in the process of lexicon enrichment, that would help not only in the research but also would solve ensure access to rare words not present in paper dictionary.

References

1. Booij, G.: Inflection and Derivation. In: Encyclopedia of Language and Linguistics. vol. 5, Elsevier, Amsterdam, London, 654-661 (2006)
2. Petic, M.: Automatic derivational morphology contribution to Romanian lexical acquisition. In: Gelbukh Al. (ed) Special issue: Natural Language Processing and its Application. Research in Computing Science, vol. 46, Mexico, 67-78 (2010)
3. Carota, F.: Derivational Morphology of Italian: Principles of Formalization. In: Literary and Linguistic Computing, Vol. 21, Suppl. Issue, 41-53 (2006)
4. Boian, E., Danilchenco A., Topal L.: The automation of speech part inflexion process. In: Computer Science Journal of Moldova, vol.1, no.2(2), Chişinău, 14-47 (1993)
5. Petic, M.: Generative models in automatic derivational morphology. In: Conference Mathematics and Information technologies: research and education (MITRE – 2011), Chişinău, Moldova, 21 - 23 august, 2011, Abstracts, Chişinău, 146-147 (2011)
6. Cojocaru, S., Boian, E., Petic, M.: Stages in automatic derivational morphology processing. In: Knowledge Engineering, Principles and Techniques, KEPT2009, Selected Papers, Cluj-Napoca, July 2 – 4, 2009, Cluj-Napoca, 97-104 (2009)

Using Natural Language Technology to Measure Mass Media Reactions in the Election Context

Daniela Gîfu[1], Dan Cristea[1, 2]

[1] "Alexandru Ioan Cuza" University, Faculty of Computer Science,
16, General Berthelot St., 700483, Iași
{daniela.gifu, dcristea}@info.uaic.ro
[2] Institute for Theoretical Computer Science, Romanian Academy – Iași branch, 2, T. Codrescu St., 700481, Iași

Abstract. This paper presents a computational tool, PEDANT, based on natural language processing (NLP) techniques for the interpretation of the political discourse in the print media. This application considers the 2009 presidential campaign in Romania. The concept behind this method is that the manner in which individuals speak and write, with the aim to deliver a certain image to the public, is an opened window towards their emotional and cognitive worlds. In other words, the vocabulary betrays the author's sensibility. By emphasizing the emotional component at the level of discourse, voters identify with the speaker, who becomes the personification of their common ideals. Our investigation is intended to give support to researchers, specialists in political sciences, to journalists and election's staff, being helpful mainly in their exploration of the political campaigns, in their intend to measure reactions with respect to the developments in the political scene.

Keywords: political discourse, natural language processing, elections, newspaper, semantic analysis, journalist.

1 Introduction

It is known that the text of the press may be of inconvenience or can be useful to those that become the subject of the press. Most of the time the subjects are the ones in political power of that time [6].

The motivation for our study relies on the need for objectivity in the interpretation of the political language, situated at the intersection of three important symbolic spaces: the political space, the public space and the communicational space, as well as on the need to measure to what extend a discourse can influence its direct receptor, the electorate and in what ways [5].

Among many attributes the political discourse has, we were interested in the lexicon and its interpretation in a range of semantic coordinates. The final objective of our research is to develop a computational framework able to offer to researchers in Mass media, political sciences, to political analysts, to the public at large (interested to consolidate their options before elections), and, why not, even to politicians themselves, the possibility to measure different parameters of a written political discourse. Based on these parameters they should be able to appreciate certain aspects characterizing the author of the discourse, as shading lights on his / her personality or on the way he / she perceives the society or only some levels of it, as well as on his/her persuasive arsenal etc. This can be done provided we will be able to link the statistical values outputted by the computational tool onto facts about the author of the message or the reality he / she is depicting. We were aware that the interpretation of numerical findings the program outputs should be validated by human experts in order to become facts. Part of our research, as reported in [2], [3] was concentrated on this type of human validation.

The software we developed, PEDANT (*Political and Emotional Discourse Analysis Tool*) offers the possibility to analyze efficiently large bodies of text and to characterize them quantitatively and qualitatively, the results having to be as close as possible to the analysis made by a human expert. The system offers a global perspective over the political discourse, as well as a punctual one.

The paper is structured as follows. Section 2 shortly describes the functionality of the software and the associated resources for the Romanian language. Then, section 3 discusses one example picked up from the print media (the editorials) during the 2009 Romanian presidential elections; section 4 highlights several challenges and sections 5 presents the conclusions.

2 The Software and Resources

The functionality of PEDANT is inspired by LIWC [4], there are important differences between the two platforms. LIWC-2007[7] is basically counting words and incrementing counters of all their declared semantic classes, when they are discovered in the input text. In the lexicon, words can be given by their long form, as a complete string of characters, or abbreviated, in which case the sign "*" plays the role of the universal jolly-joker, replacing any character. For each text in the input, LIWC produces a set of tables, each displaying the occurrences of the word-like instances of the semantic classes defined in the lexicon, as sub-unitary values. For one semantic class, such a value is computed as the number of occurrences of the words corresponding to that class divided by the total number of words in the text. It remains in the hands of the user to interpret these figures. And there is no support for considering lexical expressions.

We will refer now to the way in which PEDANT organizes the lexicon and how it counts words. The software performs part-of-speech (POS) tagging and lemmatization of words. This is why the lexicon can now be declared as a collection of lemmas having the POS categories: verb, noun, adjective and adverb. As seen, we leave out the pronouns, numerals, prepositions and conjunctions, considered to be semantically empty.

[7] Linguistic Inquiry and Word Count: www.liwc.net

An entry of the lexicon has the form: <lemma> <POS> <sem-list>, where <sem-list> is a list of semantic classes. This means that the same lemma can appear with more than one POS and, if needed, with different semantic interpretations.

Although the introduction of lemmatization and POS tagging makes useless the "any" operator, we have kept it in the definition of the lexicon entries. The user has the possibility to either define an entry as a <lemma><POS><sem-list> triple, as explained above, or as <word-root>(*) - <sem-list>, with the significance that the root can be ended with a "*" sign, no POS is defined and any word matching the root during analysis will increase the counters of all semantic classes belonging to the <sem-list>. When the PEDANT lexicon contains POS markings, the input texts have to pass a preliminary POS-tagging annotation phase before being presented to PEDANT.

The second range of differences between the two platforms stays in the user interface. In PEDANT, the user has an easy to use interface, offering a lot of services: opening of one or more files, displaying the file/s, modifying/editing and saving the text, functions of undo/redo, functions to edit the lexicon, visualizing the mentioning of instances of certain semantic classes in the text, etc. Then, the menus offer a whole range of output visualization functions, from tabular form to graphical representations and to printing services. Figure 1 shows a snapshot of the interface during a working session.

Finally, another important development was the inclusion of a collection of formulas which can be used to make comparative studies between different subjects. In section 3 we will present one example.

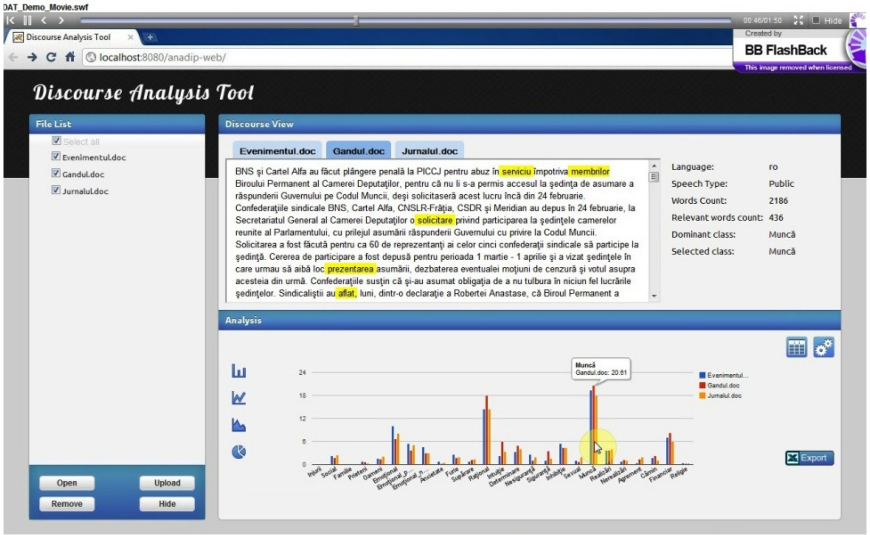
The Romanian lexicon of contains now approximately 6,000 words and roots and 29 semantic classes[8]. We plan to populate our lexicon further by importing from DEX-online[9], the greatest public online dictionary for Romanian. DEX has no semantic structure (like WORDNET, for instance). As such, an automatic assigning of semantic classes to words while importing them in the PEDANT lexicon cannot be considered for the moment (unless some sophisticated pattern matching techniques would be applied to definitions). Instead DEX-online offers a morphological tool which has been used to automatically generate all inflecting forms of the imported words and to match them against "*" - based entries. The semantic classes in PEDANT are partially placed in a hierarchy. In the future we plan to align this hierarchy with WordNet [1], for languages which support this type of linguistic resource.

A special section of the lexicon includes expressions. An expression is defined as a sequence: <root-list> => <sem-list>, in which <root-list> is a list of roots of words, therefore each optionally followed by the "*" sign. Because, in principle, a root can also be a numerical value and the semantic classes in <sem-list> are indicated by numbers, the separation between the two sections is done by the special sign (=>).

[8] These classes were selected from 64 classes (syntactic and semantic) that are included in the software LIWC-2007, original version. Taking into account the criterion of persuasion, which each stage of political discourse builds, we considered the 29 semantic classes necessary and sufficient for our study. Thus, the expected purpose will show the connection between the predominant proportion of classes and voting options. Note that the codes were changed as the American package software and class names are translated into Romanian.

[9] DEX online: www.dexonline.ro

Each time a sequence of words matching the <root-list> is recognized in the text, the counters associated with the semantic classes in the <sem-list> are increased.



BNS şi Cartel Alfa au făcut plângere penală la PICCJ pentru abuz în serviciu împotriva membrilor Biroului Permanent al Camerei Deputaţilor, pentru că nu li s-a permis accesul la şedinţa de asumare a răspunderii Guvernului pe Codul Muncii, deşi solicitaseră acest lucru încă din 24 februarie. Confederaţiile sindicale BNS, Cartel Alfa, CNSLR-Frăţia, CSDR şi Meridian au depus în 24 februarie, la Secretariatul General al Camerei Deputaţilor o solicitare privind participarea la şedinţele camerelor reunite al Parlamentului, cu prilejul asumării răspunderii Guvernului cu privire la Codul Muncii. Solicitarea a fost făcută pentru ca 60 de reprezentanţi ai celor cinci confederaţii sindicale să participe la şedinţă. Cererea de participare a fost depusă pentru perioada 1 martie - 1 aprilie şi a vizat şedinţele în care urmau să aibă loc prezentarea asumării, dezbaterea eventualei moţiuni de cenzură şi votul asupra acesteia din urmă. Confederaţiile susţin că şi-au asumat obligaţia de a nu tulbura în niciun fel lucrările şedinţelor. Sindicaliştii au aflat, luni, dintr-o declaraţie a Robertei Anastase, că Biroul Permanent a

Language: ro
Speech Type: Public
Words Count: 2186
Relevant words count: 436
Dominant class: Muncă
Selected class: Muncă

Fig. 1. The PEDANT Interface.

In Fig.1, in the left window appear the selected files, in the middle window - the text from the selected file - and in the right window, information about the text (language, speech type, word count, dominant class etc.). Bellow appears a graph selected from different types. By selecting a specific class on the top-right, in the middle window, all words assigned to that class are highlighted in the text.

The 29 semantic classes included now in PEDANT have been selected to fit optimally with the necessities to interpret the political discourse of the presidential campaign in Romania, in 2009. We were mainly interested to determine those political attitudes which were able to influence the voting decision of the electorate. However, the user can define at his / her will these classes and the associated lexicon. Here are the classes included in the PEDANT package:

```
<classes>
    <class name="Injurii" id="1"/>
    <class name="Social" id="2"/>
    <class name="Familie" id="3" parent="2"/>
    <class name="Prieteni" id="4" parent="2"/>
    <class name="Oameni" id="5" parent="2"/>
    <class name="Emoţional" id="6"/>
    <class name="Emoţional_pozitiv" id="7" parent="6"/>
    <class name="Emoţional_negativ" id="8" parent="6"/>
    <class name="Anxietate" id="9" parent="8"/>
    <class name="Furie" id="10" parent="8"/>
    <class name="Supărare" id="11" parent="8"/>
    <class name="Raţional" id="12"/>
    <class name="Intuiţie" id="13" parent="12"/>
    <class name="Determinare" id="14" parent="12"/>
    <class name="Nesiguranţă" id="15" parent="12"/>
    <class name="Siguranţă" id="16" parent="12"/>
    <class name="Inhibiţie" id="17" parent="12"/>
    <class name="Perceptiv" id="18"/>
    <class name="Vizual" id="19" parent="18"/>
    <class name="Auditiv" id="20" parent="18"/>
    <class name="Tactil" id="21" parent="18"/>
    <class name="Sexual" id="22"/>
    <class name="Muncă" id="23"/>
    <class name="Realizări" id="24"/>
    <class name="Nerealizări" id="25"/>
    <class name="Agrement" id="26"/>
    <class name="Cămin" id="27"/>
    <class name="Financiar" id="28"/>
    <class name="Religie" id="29"/>
</classes>
```

Keeping in mind the remarkably sophisticated and time consuming process in which the Romanian version of the dictionary LIWC-2007 was acquired, we knew that some decisions have to be taken in order to optimize its content while also diminishing the influence of English, evident if a simple EN-onto-RO translation process would have been applied. The development of the lexicon was done in several phases:

- first the LIWC-2007 English terms belonging to the 29 classes previously mentioned, considered meaningful for this type of analysis, were translated, retaining only the Romanian words which had senses in accordance with the corresponding classes;
- then, words / roots in each class were sorted alphabetically;
- then we have reconsidered each class in part, eliminating words that could introduce ambiguities and including synonyms. We have done this activity with a class of master students in Computational Linguistics[10]. Then, the work done by students has been once again validated by both authors. By working within one semantic class at a time, we were able to easily recognize classification mistakes and correct them. Also, the alphabetical ordering offers the possibility to operate certain roots optimizations, by exploiting the use of the jolly-joker "*";
- then we verified a part of the documents. When a competent reader does this, normally do pop-up important words. They were checked against the generated morphological variants of the DEX-online lexicon by using "*" to include more

[10] In 2010, in their 1st year at the Faculty of Computer Science, the "Alexandru Ioan Cuza" University of Iaşi.

possibilities and the resulting matches were included in the PEDANT-2011 lexicon.

- then, the dictionaries assigned to all classes were merged, and the obtained list was sorted again alphabetically. Multiple occurrences were removed (by clashing together the lists of classes, and leaving only one instance of a class in the list corresponding to one root/word);
- then, when the root/word notation (including or not the use of "*") was seen to give rise to unwanted ambiguities, the <lemma><POS> notation was used instead;
- finally, starting from words, expressions were introduced;
- as mentioned already, in a future stage, we will use the words we have now in the lexicon as seeds in a trial to enrich it automatically, by making use of DEX-online. We study now strategies to exploit the synonymy relation and the definitions.

3 The Analysis of Print Media Discourse During the 2009 Romanian Presidential Elections

For the elaboration of preliminary conclusions over the presidential elections process, conducted in the period October – December 2009 in Romania, we collected, stored and processed electronically, in three different stages (one month before, the 1st tour and, finally, the 2nd tour of the election's campaign), political texts, i.e. editorials published by three national publications having similar[11] profiles.

The monitored written media corpus was processed directly with PEDANT. The speech records were previously manually transposed onto text and then they followed the same processing as the written texts. In essence, the program receives the input from one or more text files, and counts occurrences of words belonging to its defined classes. The user can notice directly the mentioned semantic classes (and the corresponding frequencies), as the words belonging to a selected class appear highlighted in the selected text. The user can choose a type of graphical representation, which gives intuitive visual perceptions on which the interpretation of discourse data can be performed more conveniently.

Apart from simply computing frequencies, the system can also perform comparative studies. The assessments made are comprehensive over the selected classes because they represent averages on collections of texts, not just a single text.

PEDANT provides a library of comparative functions, with 2 to *n* different input streams of data. One stream can be either a newspaper, or only one discursive approach on a certain topic delivered at a certain moment in time by the traced author. To exemplify, one type of graphics considered for the interpretation was the one-to-

[11] The three newspapers have been *Evenimentul Zilei*, *Gandul* and *Ziua* (www.mediapres.ro), which are known to have a common profile: national dailies of general information, tabloids with a circulation of tens of thousands of copies per edition, each. The newspapers were monitored on their websites: *Evenimentul zilei* – www.evz.ro, *Gândul* – www.gandul.info, *Ziua* – www.ziua.ro.

two difference, as given by Formula (1), included in the PEDANT Mathematical Functions Library:

$$Diff_{x,y,z}^{1-2} = average(x) - \frac{average(y) + average(z)}{2} \qquad \textbf{(1)}$$

where *x*, *y* and *z* are three streams; *average*(*x*), *average*(*y*) and *average*(*z*) are the average frequencies of *x*, *y* and *z* over the whole stream, and the difference is computed for each selected class. Since a difference can lead to both positive and negative values, these particular graphs should read as follows: values above the horizontal axis are those prevailing at the daily *x* versus the daily *y* and *z*, and those below the horizontal axis show the reverse prominence. A zero value indicates equality.

To exemplify, we present below a chart with two streams of data, representing the editorials of the second tour of voting.

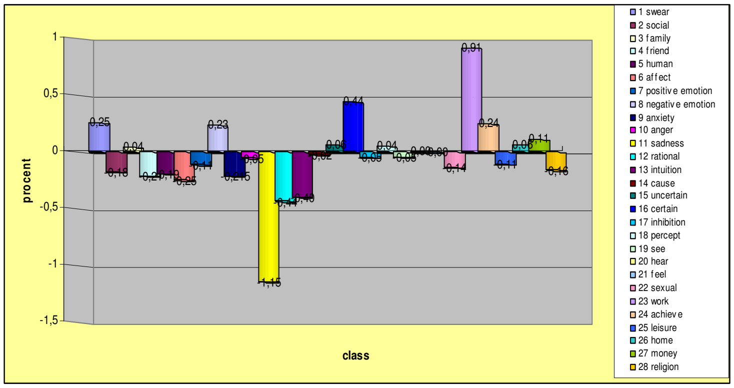


Fig. 2. The average differences in the frequencies for each class after processing the editorials (the second tour), between the newspapers: *Evenimentul Zilei* versus *Gândul* & *Ziua*.

Our experience shows that an absolute difference value below the threshold of 0.5% should be considered as irrelevant and, therefore, ignored in the interpretation. So, the graphical representation in Fig. 2, in which *Evenimentul zilei* is compared with *Gândul* & *Ziua*, should be interpreted as follows: *Evenimentul zilei*'s discursive intervention has an interest more towards the working aspects (`Work`) than *Gândul* and *Ziua* together, which had a negative emotional (`Sadness`) attitude.

Many of the conclusions found by the program have been confirmed by political commenter's. Moreover, the program helped also to outline distinctive features which brought a new, and sometimes even unexpected, vision upon the discursive characteristics of the political locator, of the columnists and, last but not least, of the Romanian voters, at the end of 2009.

4 Discussion: Challenges

The first challenge arises from interpretation of what constitutes persuasion. In general, journalists/political candidates performed better when their themes were treated with excessive emotional tonalities. In this case, the lectors can be manipulated easily (e.g., Sadness class). It would be beneficial if humans could defer to a machine learning prediction in cases where they lack confidence.

Second, our data elicitation task was opened, allowing all analysts, journalists, communication specialists to make choices and us to present their discourse preferences depending on their interests. The heterogeneous data, however, lessens linguistic predictive power. Previous computational attempts succeeded with well-defined tasks for discovering work aspects (as given by the Work class). Also, following the qualitative interpretation of different types of emotion levels, in the future we will analyze the election context in function to the public agenda versus the political candidate agenda. We are aware that many technological aspects have yet to be refined and enhanced. One of the most important is the determination of the senses of words and expressions in context. The negations will also be considered, as the semantic class a word-in-context belongs to. Another line to be continued regards the evaluation metrics, which have not received enough attention till now. We are currently studying other statistical metrics able to give a more comprehensive image on different facets of the political discourse.

5 Conclusions

We believe that PEDANT has a range of features that make it attractive as a tool to assist political campaigns. It can also be rapidly adapted to new domains and to new languages, while its interface is user-friendly and offers a good range of useful functionalities.

In the future we intend to include a word sense disambiguation module in order to determine the correct senses, in context, of those words which are ambiguous between different semantic classes belonging to the lexicon, or between classes in the lexicon and outside the lexicon (in which case they would not have to be counted).

Acknowledgments. This paper is a result of the project “Transnational Network for Integrated Management of Postdoctoral Research in Communicating Sciences. Institutional building (postdoctoral school) and fellowships program (CommScie)” - POSDRU/89/1.5/S/63663, financed under the Sectoral Operational Program Human Resources Development 2007-2013. The PEDANT software has been developed by Mădălina Spătaru from the Faculty of Computer Science “Alexandru Ioan Cuza” of Iași. The master students in the 1st year of Computational Linguistics at the Faculty of Computer Science of the “Alexandru Ioan Cuza” of Iași have participated in the initial phases of the acquisition of the lexicon.

References

1. Fellbaum, Ch. (ed.).: WordNet, An Electronic Lexical Database. The MIT Press (2001)
2. Gîfu, D., Cristea, D.: Computational Techniques in Political Language Processing: AnaDiP-2011, In J. J. Park, L. T. Yang, and C. Lee (Eds.): FutureTech 2011, Part II, CCIS 185, 188–195 (2011).
3. Gîfu, D.: The Discourse of the Written Press and the Violence of Symbols (in Romanian), PhD thesis, Faculty of Philosophy and Political Studies, "Alexandru Ioan Cuza" University of Iași (2010)
4. Pennebaker, J. W., Francis, Martha E., Booth, R. J.: Linguistic Inquiry and Word Count – LIWC2001, Mahwah, NJ, Erlbaum Publishers (2001)
5. Perlmutter, D. D.: The Manship School guide to political communication, Baton Rouge, Louisiana State University Press (1999)
6. Touraine, A.: Critique de la modernité, Fayard, Paris, (1992)

Using Textual Entailment in Internet Surveillance

Adrian Iftene

"Al. I. Cuza" University of Iasi, Faculty of Computer Science
adiftene@info.uaic.ro

Abstract. This paper focuses on some concepts related to surveillance of users and information. The purpose of our application is to create clusters with fragments of text for user opinions regarding certain products or events. In order to create clusters we use a Textual Entailment system which allows us to identify relations between fragment of texts like entailment, contradiction or unknown. These clusters will help us in the process of analyzing of positive, negative and neutral opinions, depending on what interest us. One of the focuses of our work is helping companies build such analysis in the context of users' sentiment identification. Therefore, the corpus we work on consists of articles of newspapers, blogs, various entries of forums, and posts in social networks. Later based on analysis of existing opinions from texts the companies can establish new trends and policies and determines in which areas investments must be made.

Keywords: Textual Entailment, Internet Surveillance, Sentiment Analysis.

1 Introduction

In the past years, new user communities appear over the Internet, like personal websites where users have relations like friendship or kinship (see for example Facebook[12], Myspace[13], Twitter[14]), or professional websites where users have business-related connections (see LinkedIn[15]). In these communities, users freely express their opinions about common topics, critic or approve certain aspects related to their common topic. What is interesting is the fact that, for example, in case of a product, based on user opinions, we can have an overview of the product quality; we can identify the advantages of using this product and even its weaknesses.

This paper focuses on some concepts related to surveillance of users and information. The purpose of our application is to find out user opinions regarding certain products or events and to group them on positive, negative or neutral opinions. One of the components of our system is based on a Textual Entailment system which is able

[12] Facebook: http://www.facebook.com/
[13] MySpace: http://www.myspace.com/
[14] Twitter: http://twitter.com/
[15] Linkedin: http://www.linkedin.com/

to identify relations between fragments of text like entailment, contradiction or unknown.

In the first part of this paper we present the main concepts used to develop our system: Internet Surveillance, Sentiment Analysis and Textual Entailment. Next we offer the details of the system build in order to get what we want. In the last part we present the rules (positive and negative) to show how our application works: we will find out how we can group users opinions regarding their common interest.

2 Internet Surveillance

The *surveillance* means the accumulation of information defined as symbolic materials that can be stored by an agency or community as well as the supervision of the activities of subordinates by their superiors within any community [1]. The modern nation state was from its beginning an information society, because it collects and stores information about citizens (births, marriages, deaths, demographic and fiscal statistics, “moral statistics” relating to suicide, divorce, delinquency, etc.) in order to organize the administration.

Internet surveillance is related to information surveillance over the Internet and it is done for different reasons than those related to national security (especially after the 11th of September 2001 terrorist attack) and marketing interest for big companies [2]. Nowadays, big companies and organizations spend time and money in order to find users’ opinions about their products, the impact of their marketing decisions, or the overall feeling about their support and maintenance services. This analysis helps in the process of establishing new trends, policies and determines in which areas investments must be made.

In order to identify users’ opinion on certain products, we have created several modules that perform Internet surveillance, aiming at extracting new useful information for our application. Therefore, we considered the results obtained after searching the Internet using Google search engine, filtered to newspapers, forums and blogs, but also we keep our eyes on social networks like Twitter.

For this part we build a crawling component able to extract relevant information’ related to some keywords. Also, this component is able to monitor a list of RSS and to extract all new pages from there.

3 Sentiment Analysis

Sentiment analysis, i.e. the analysis and classification of the opinion expressed by a text on its subject matter, is a form of information extraction from text, which recently focused a lot of research and growing commercial interest. Using Sentimatrix [3], a sentiment analysis service, we seek to explore how sentiment analysis methods perform across languages.

Starting from the early 1990s, the research on sentiment-analysis and point of views generally assumed the existence of sub-systems for rather sophisticated NLP tasks, ranging from parsing to the resolution of pragmatic ambiguities [4, 5, 6]. In Sentimatrix [3], in order to identify the sentiment a user expresses about a specific product or company, the company name must be first identified in the text. Named entity recognition (NER) systems typically use linguistic grammar-based techniques or statistical models (an overview is presented in [7]). Hand-crafted grammar-based systems typically obtain better precision, but at the cost of lower recall and months of work by experienced computational linguists. Besides, the task is hard to adapt to new domains. Various sentiment types and levels have been considered, starting from the "universal" six levels of emotions considered in [8, 9, 10]: anger, disgust, fear, happiness, sadness, and surprise. For Sentimatrix, we adapted this approach to five levels of sentiments: strong positive, positive, neutral, negative and strong negative.

The first known systems relied on relatively shallow analysis based on manually built discriminative word lexicons [11], used to classify a text unit by trigger terms or phrases contained in a lexicon. The lack of sufficient amounts of sentiment annotated corpora led the researchers to incorporate learning components into their sentiment analysis tools, usually supervised classification modules, (e.g., categorization according to affect), as initiated in [12].

Much of the literature on sentiment analysis has focused on text written in English. Sentimatrix is designed to be, as much as possible, language independent, the resources used being easily adaptable for any language.

4 Textual Entailment

While the language variability problem is well known in Computational Linguistics, a general unifying framework has been proposed only in last year's [13]. In this approach, language variability is addressed by defining the notion of entailment as a relation that holds between two language expressions (i.e. a text T and a hypothesis H) if the meaning of H, as interpreted in the context of T, can be inferred from the meaning of T. The entailment relation is directional as the meaning of one expression can entail the meaning of the other, while the opposite may not.

Recognizing Textual Entailment (RTE) was proposed in 2005 as a generic task, aimed at building systems capable of capturing the semantic variability of texts and performing natural language inferences. These systems can be integrated in NLP applications, improving their performance in fine-grained semantic analysis.

The first important direction from RTE-1 was an application of the BLEU algorithm for recognising textual entailments. The BLEU[16] algorithm was created by [14] as a procedure to rank systems according to how well they translate texts from one language to another. Basically, the algorithm looks for *n-gram* coincidences between a candidate text (the automatically produced translation) and a set of reference texts (the human-made translations). One of the first approaches based on BLEU algorithm [15] consists in using the BLEU algorithm that works at the lexical level, to compare the entailing text (T) and the hypothesis (H). Next, the entailment is judged as true or false according to BLEU's output.

[16] Bilingual Evaluation Understudy

Graph distance/similarity measures are widely recognized to be powerful tools for matching problems in computer vision and pattern recognition applications [16] and it was used with success in RTE of 2005 by Pazienza, Pennacchiotti and Zanzotto. Objects to be matched (two images, patterns, text and hypothesis in RTE task, etc.) are represented as graphs, turning the recognition problem into a graph matching task. Thus, following [13], since the hypothesis *H* and text *T* may be represented by two syntactic graphs, the textual entailment recognition problem can be reduced to graph similarity measure estimation, although textual entailment has particular properties [17].

The core of approach presented in [18] is a tree edit distance algorithm applied on the dependency trees of both the text and the hypothesis. If the distance (i.e. the cost of the editing operations) among the two trees is below a certain threshold, empirically estimated on the training data, then the authors assign an entailment relation between the two texts. The authors designed a system based on the intuition that the probability of an entailment relation between *T* and *H* is related to the ability to show that the whole content of *H* can be mapped into the content of *T*. The next systems were more complex and combined the initial approach with machine learning algorithms [19] or used the probabilistic transformations of the trees [20]. Other systems used the representation of texts with dependency trees using MINIPAR[17] for that [21, 22, 23].

In the first edition of 2005, five groups used logical provers and offered deep semantic analysis. One system [24] transformed the text and hypothesis into logical formula (like in [25]) and it calculated the “cost” of proving hypothesis from text. In 2006 only two systems used logical inferences and one of the systems achieved the second result of the edition [26]. In 2007 the number of systems using logical inferences grew up to seven and the first two results used the logical inferences [27, 28]. In RTE-4 nine groups used logical inferences in order to identify the entailment relation, and two of them were oriented to that [29, 30].

Starting with RTE-2, the interest for using machine learning grew constantly. Thus, the number of systems that used machine learning for classification was increased from seven in 2005 to fifteen in 2006 and sixteen in 2007 and 2008. The approaches are various and their results depend on the capability of authors to identify relevant features. In [31], matching features are represented by lexical matches (including synonyms and related words), part-of-speech matching and matching of grammatical dependency relations. Mismatch features include negation and numeric mismatches. The MLEnt system [32] models lexical and semantic information in the form of attributes and, based on them, proposed 17 features. In [33], the authors computed a set of semantic based distances between sentences. The system of [34] used semantic distance between stems, subsequences of consecutive stems and trigrams matching. The features identified by [35] include lexical semantic similarity, named entities, dependent content word pairs, average distance, negation, and text length.

[17] MINIPAR: http://ai.stanford.edu/~rion/parsing/minipar_viz.html

5 System Components

The architecture and the main modules of our system are: preprocessing (Segmenter, Tokenizer, and Language Detector), named entity extraction, anaphora resolution, opinion identification and textual entailment. Each module of the system can be exposed in a user-friendly interface. The final production system is based on service oriented architecture in order to allow users flexible customization and to enable an easier way for marketing technology.

5.1 Sentiment Analysis Module

In such a task as sentiment identification, linguistic resources play a very important role [3]. The core resource is a manually built **list of words and groups of words** that semantically signal a positive or a negative sentiment. From now on, we will refer to such a word or group of words as "**sentiment trigger**". Certain weights have been assigned to these words after multiple revisions. The weights vary from -3, meaning strong negative to +3, which translates to a strong positive. There are a total of 3,741 sentiment triggers distributed to weight groups. The triggers are lemmas, so the real number of words that can be identified as having a sentiment value is much higher.

This list is not closed and it suffers modifications, especially by adding new triggers, but in certain cases, if a bad behaviour is observed, the weights may also be altered.

We define a **modifier** as a word or a group of words that can increase or diminish the intensity of a sentiment trigger. We have a manually built list of modifiers. We consider negation words a special case of modifiers that usually have a greater impact on sentiment triggers. So, we also built a small **list of negation words**.

Global sentiment computation: In this section, we will describe how the cumulative value of a sentiment segment is computed.

At the base of a sentiment segment stands the given weight of the sentiment trigger that is part of the general segment. Besides that, modifiers and negation words have a big impact. For example, consider the following three sentences.

1. *John is a good person.*
2. *John is a very good person.*
3. *John is the best.*

In the first one, a positive sentiment is expressed towards *John*. In the second one, we also have a positive sentiment, but it has a bigger power and in the third one the sentiment has the strongest intensity.

We distinguish two separate cases in which negation appears. The first one is when the negation word is associated with a sentiment trigger and it changes a positive one into a negative trigger and vice versa; and the second one refers to the case in which the negation affects a trigger accompanied by a modifier. We illustrate these situations in the following examples.

A1. *John is a good person.*

A2. *John is not a good person.*
B1. *John is the best.*
B2. *John is not the best.*

If we assign the weight +2 to *good* in the A1 sentence, it is safe to say that in A2, *not good* will have the weight -2. From a semantic perspective, we have the **antonym relation** (between *good* and *not good*) and the **synonym relation** (between *not good* and *bad*).

On the other hand, in the B2 example, *not the best* is not the same as *the worst*, the antonym of *the best*. In this case, we consider *not the best* to be somewhere between *good* and *the best*.

5.2 Textual Entailment Component

The main idea of the system built for the competition in 2007 [22] and improved for the competition in 2008 [36] is to map every node from the hypothesis to a node from the text using extensive semantic knowledge from sources like DIRT, WordNet, Wikipedia, VerbOcean and acronyms database. After the mapping process, we associate a local fitness value to every word from the hypothesis, which is used to calculate a global fitness value for current fragments of text. The global fitness value is decreased in cases in which a word from the hypothesis cannot be mapped to one word from the text or when we have different forms of negations for mapped verbs. In the end, using thresholds identified in the training step for global fitness values, we decide, for every pair from test data, whenever the entailment holds.

The systems with which we participated in RTE-4, RTE-5 [37] and RTE-6 [38] represent enhanced versions of the systems used in RTE-3 [22] and in RTE-4 [36]. Additionally, we added new rules and used new semantic resources with the aim to better identify the contradiction cases.

In order to improve the quality of tools output, before using them on the initial test file some additional steps are performed. Thus, we replace in all test data the expression “hasn’t” with “has not”, “isn’t” with “is not”, etc. The meaning of the text remains the same after transformation, but the MINIPAR output is better for this new text. Also, before sending the text to the LingPipe, we replace in the initial test texts some punctuation signs like quotation marks “”, brackets (), [], {}, commas, etc. with the initial sign between spaces. Again, the meaning of the text is the same, but the LingPipe output is better processed further after this transformation.

After the preprocessing steps, all files obtained are sent to the LingPipe[18] module and we used gazetteer from GATE [39] in order to find the named entities. Along with the identification of named entities, we transform both the text and the hypothesis into dependency trees with MINIPAR[19] [40]. The output produced by MINIPAR is a graph in which the nodes are the words for the parsed sentence labelled with their grammatical categories and the edges are relations between the words labelled with grammatical relationships.

[18] LingPipe: http://alias-i.com/lingpipe/
[19] MINIPAR: http://www.cs.ualberta.ca/~lindek/minipar.htm

From now on, the main goal is to map every entity in the dependency tree associated with the hypothesis (called the *hypothesis tree*) to an entity in the dependency tree associated with the text (called the *text tree*) [22].

The mapping between entities can be negotiated in two ways: *directly* (when entities from hypothesis tree exist in the text tree) or *indirectly* (when entities from text tree or hypothesis tree cannot be mapped directly and need transformations using external resources). Using this type of mapping between an entity from hypothesis tree and an entity from text tree, we calculate a *local fitness value* which indicates the appropriateness between entities. Based on local fitness of node and on local fitness of its father, we build for node an *extended local fitness* and in the end, using all partial values, we calculate a normalized value that represents the *global fitness*.

When an entity belonging to the hypothesis tree can be mapped directly to more entities from the text tree, we select the mapping which increases the global fitness with the highest value. When it is not possible to map an entity of the hypothesis tree to another entity of the text tree, we use external resources for that: DIRT, VerbOcean and WordNet (for verbs), Acronyms database and Background Knowledge (for named entities), eXtended WordNet and WordNet (for nouns and adjectives).

6 Rules used in Opinions Classification

Some components of our textual entailment system were adapted in order to be used in the process of opinions and feelings classification. Our assumption was that the Textual Entailment component can help a system based on sentiment analysis services in order to improve this accuracy in order to identify positive or negative user's opinions.

In bellow Figure we can see how the interface of Sentimatrix application displayed the statistics calculated using two clusters of users' opinion: positive and negative.

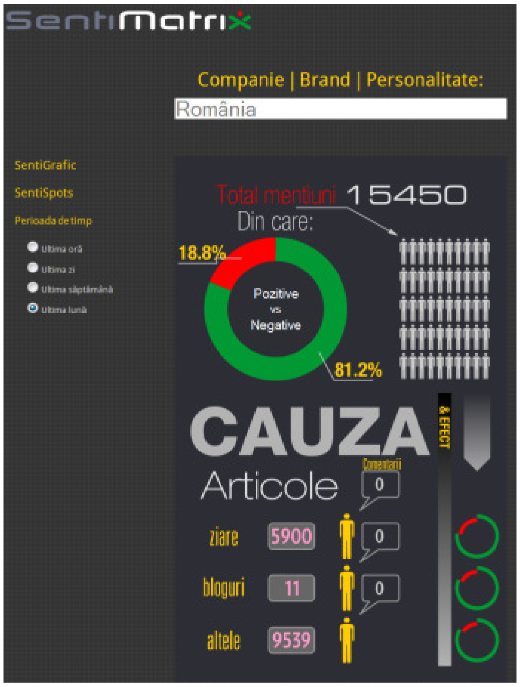
To obtain clusters of opinions we create special rules for identifying entailment relations between fragments of text. The rules will help us to decide if we must put the texts in the same cluster or in different clusters.

6.1 Positive Rules

In this case any type of mapping between two fragments of text will increase the global score and, in the end, will increase the probability to have the texts with the same associated value for opinions. For every node of the hypothesis tree which can be mapped directly to a node of the text tree, we will consider the local fitness value to be 1 (the maximum value).

When it is not possible to do a direct mapping between a hypothesis node and a text node, we will try to use the external resources and to transform the hypothesis node in an equivalent one.

Thus, for *verbs* we use the DIRT resource [40] and transform the hypothesis tree into an equivalent one, with the same nodes, except the verb. If the word is a *named* *entity*, we try to use a database of acronyms[20] or obtain information related to it from the background knowledge [41]. As an example for acronyms we can indicate the acronyms of cities (NY is New York, LA is Los Angeles) of companies (GDF is Gaz de France, BP is British Petroleum) of parties (RLP is Republican Labour Party, SDP Social Democratic Party). From Wikipedia we extract pairs of named entities with following types of relations: *is_similar_with* (Orange *is_similar_with* Orange Romania), *is_included_in* (Bucharest *is_included_in* Romania), etc.



Companie | Brand | Personalitate: România

Perioada de timp
○ Ultima oră
○ Ultima zi
○ Ultima săptămână
● Ultima lună

Fig. 1. Users' Opinions Related to Country "Romania".

For *nouns* and *adjectives* we use WordNet [42] and a part of the relations from the eXtended WordNet[21] to look up synonyms and then we try to add these words in our resources with sentiment triggers. We only took relations with high scores of similarity from eXtended WordNet. Example in which the synonymy relation as given by WordNet is used is: relation between *poor* and *insufficient* and *inadequate*), etc.

[20] Acronym Guide: http://www.acronym-guide.com

[21] eXtended WordNet: http://xwn.hlt.utdallas.edu/

6.2 Negative Rules

For every verb from the hypothesis a Boolean value is considered, which indicates whether the verb is negated or not. To find that, we check inside its tree on its descending branches to see whether one or more of the following words can be found: *not*, *never*, *may*, *might*, *cannot*, etc. For each of these words the initial truth value associated with the verb is successively negated, which by default is "false". The final value depends on the number of such words.

Another rule was built for the particle "to" when it precedes a verb. In this case, the sense of the infinitive is strongly influenced by the active verb, adverb or noun before the particle "to", as follows: if it is being preceded by a verb like *believe*, *glad*, *claim* or their synonyms, or adjective like *necessary*, *compulsory*, *free* or their synonyms, or nouns like *attempt*, *trial* and their synonyms, the meaning of the verb in the infinitive form is stressed upon and becomes "certain". For all other cases, the "to" particle diminishes the certainty of the action expressed in the infinitive-form verb.

7 Conclusions

In this paper we present the main components of our system: the *crawling component* (responsible with the extracting of relevant information from the Internet), the *sentiment analysis component* (which is able with opinion identification) and the *textual entailment component* (which is able with grouping of users' opinions).

The sentiment analysis component uses resources with triggers words that signal the existence of users' opinions, resources with modifiers that amplifies or diminishes the valences of sentiment triggers and resources with resources with negation words.

The textual entailment component groups the users' opinions in clusters with positive, negative and neutral opinions. This component uses *positive rules*, based on using of resources like DIRT, acronyms databases, WordNet, ontologies, and *negative rules*, based on identification of negations and of contexts that change the verbs sense.

The main problems of our system are related to cases when we try to see the entailment relation between sentences (text and hypothesis), and both sentences are without sentiment triggers.

Acknowledgments. The research presented in this paper is funded by the Sectoral Operational Program for Human Resources Development through the project "Development of the innovation capacity and increasing of the research impact through post-doctoral programs" POSDRU/89/1.5/S/49944.

References

1. Giddens A.: A contemporary critique of Historical Materialism, Vol. 1: Power, property and the state, Macmillan, London, (1981)
2. Fuchs C.: Social Networking Sites and the Surveillance Society. A Critical Case Study of the Usage of studiVZ, Facebook, and MySpace by Students in Salzburg in the Context of

Electronic Surveillance, Salzburg/Vienna, Austria, 2009; Forschungsgruppe “Unified Theory of Information” - Verein zur Förderung der Integration der Informationswissenschaften, ISBN 978-3-200-01428-2 (2009)

3. Gînscă, A. L., Boroş, E., Iftene, A., Trandabăţ, D., Toader, M., Corîci, M., Perez, C. A., Cristea, D. Sentimatrix - Multilingual Sentiment Analysis Service. In Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (ACL-WASSA2011) Portland, Oregon, USA, June 19-24 (2011)
4. Hearst, M.: Direction-based text interpretation as an information access refinement. In Paul Jacobs, editor, Text-Based Intelligent Systems, Lawrence Erlbaum Associates, 157-274 (1992)
5. Wiebe, J. M.: Identifying subjective characters in narrative. In Proceedings of the International Conference on Computational Linguistics (COLING), 401–408 (1990)
6. Wiebe, J. M.: Tracking point of view in narrative. Computational Linguistics, 20(2), 233–287 (1994)
7. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification, Linguisticae Investigationes 30, no. 1, Publisher: John Benjamin’s Publishing Company (2007) 3-26
8. Ovesdotter Alm, C., Roth, D., Sproat, R.: Emotions from text: machine learning for text-based emotion prediction. In Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP) (2005)
9. Liu, H., Lieberman, H., Selker, T.: A model of textual affect sensing using real-world knowledge. In Proceedings of Intelligent User Interfaces (IUI), 125–132 (2003)
10. Subasic, P., Huettner, A.: Affect analysis of text using fuzzy semantic typing. IEEE Transactions on Fuzzy Systems, 9(4), 483–496 (2001)
11. Tong, R. M.: An operational system for detecting and tracking opinions in on-line discussion. In Proceedings of the Workshop on Operational Text Classification (OTC) (2001)
12. Wiebe, J., Bruce, R.: Probabilistic classifiers for tracking point of view. In Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation, 181–187 (1995)
13. Dagan, I. and Glickman, O.: Probabilistic textual entailment: Generic applied modeling of language variability. In Learning Methods for Text Understanding and Mining, Grenoble, France (2004)
14. Papineni, K., Roukos, S., Ward, T., Zhu, W.: BLEU: a method for automatic evaluation of machine translation. Research report, IBM (2001)
15. Pérez, D., Alfonseca, E.: Application of the Bleu algorithm for recognising textual entailments. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, 11–13 April, Southampton, U.K., 9-12 (2005)
16. Pazienza, M.T., Pennacchiotti, M., Zanzotto, F. M.: Textual Entailment as Syntactic Graph Distance: a rule based and a SVM based approach. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, 25–28 April, Southampton, U.K. (2005)
17. Bunke, H. and Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recogn. Lett., 19 (3-4), 255–259 (1998)
18. Kouylekov, M. and Magnini, B.: Recognising Textual Entailment with Tree Edit Distance Algorithms. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, 25–28 April, Southampton, U.K., 17-20 (2005)
19. Kozareva, Z., Montoyo, A.: MLEnt: The Machine Learning Entailment System of the University of Alicante. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, 10 April, Venice, Italia, 17-20 (2006)

20. Harmeling, S.: An Extensible Probabilistic Transformation-based Approach to the Third Recognising Textual Entailment Challenge. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. 28-29 June, Prague, Czech Republic, 137-142 (2007)
21. Katrenko, S. and Adriaans, P.: Using Maximal Embedded Syntactic Subtrees for Textual Entailment Recognition. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, 10 April, Venice, Italia, 33-37 (2006)
22. Iftene, A., Balahur-Dobrescu, A.: Hypothesis Transformation and Semantic Variability Rules Used in Recognising Textual Entailment. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. 28-29 June, Prague, Czech Republic, 125-130 (2007)
23. Bar-Haim, R., Berant, J., Dagan, I., Greental, I., Mirkin, S., Shnarch, E., Szpektor, I.: Efficient Semantic Deduction and Approximate Matching over Compact Parse Forests. In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, Gaithersburg, Maryland, USA (2008)
24. Raina, R., Haghighi, A., Cox, C., Finkel, J., Michels, J., Toutanova, K., MacCartney, B.,, Marneffe, M.C., Manning, C., Ng, A., Y.: Robust Textual Inference using Diverse Knowledge Sources. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, 11–13 April, 2005, Southampton, U.K., 57-60 (2005)
25. Harabagiu, S., Pasca, M., Maiorano, S.: Experiments with Open-Domain Textual Question Answering. COLING 2000 (2000)
26. Tatu, M., Iles, B., Slavick, J., Novischi, A., Moldovan, D.: COGEX at the Second Recognising Textual Entailment Challenge. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, 10 April, 2006, Venice, Italy, 17-20 (2006)
27. Hickl, A., Bensley, J.: A Discourse Commitment-Based Framework for Recognising Textual Entailment. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. 28-29 June, Prague, Czech Republic, 185-190 (2007)
28. Tatu, M., Moldovan, D.: COGEX at RTE3. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. 28-29 June, Prague, Czech Republic, 22-27 (2007)
29. Clark, P., Harrison, P.: Recognizing Textual Entailment with Logical Inference. In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, Gaithersburg, Maryland, USA (2008)
30. Bergmair, R.: Monte Carlo Semantics: MCPIET at RTE4. In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, 2008. Gaithersburg, Maryland, USA (2008)
31. Inkpen, D., Kipp, D., Nastase, V.: Machine Learning Experiments for Textual Entailment. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, 10 April, Venice, Italy, 17-20 (2006)
32. Li, B., Irwin, J., Garcia, E.V., Ram, A.: Machine Learning Based Semantic Inference: Experiments and Observations at RTE-3. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. 28-29 June, Prague, Czech Republic (2007) 159-164
33. Kozareva, Z. and Montoyo, A.: MLEnt: The Machine Learning Entailment System of the University of Alicante. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, 10 April, Venice, Italy, 17-20 (2006)
34. Ferrés, D., Rodríguez, H.: Machine Learning with Semantic-Based Distances Between Sentences for Textual Entailment. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. 28-29 June, Prague, Czech Republic (2007) 60-65

35. Montejo-Ráez, A., Perea, J.M, Martínez-Santiago, F., García-Cumbreras, M. Á., Martín-Valdivia, M., Ureña-López, A.: Combining Lexical-Syntactic Information with Machine Learning for Recognising Textual Entailment. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. 28-29 June, Prague, Czech Republic (2007) 78-82
36. Iftene, A.: UAIC Participation at RTE4. In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, Gaithersburg, Maryland, USA (2008)
37. Iftene, A., Moruz, A.M.: UAIC Participation at RTE5. In Text Analysis Conference (TAC 2009) Workshop - RTE-5 Track. National Institute of Standards and Technology (NIST). November 16-17, Gaithersburg, Maryland, USA (2009)
38. Iftene, A., Moruz, M.A.: UAIC Participation at RTE-6. In Text Analysis Conference (TAC 2010) Workshop - RTE-6 Track. National Institute of Standards and Technology (NIST). November 15-16, 2010. Gaithersburg, Maryland, USA (2010)
39. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: an architecture for development of robust HLT applications. In ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Morristown, NJ, USA (2001) 168-175
40. Lin, D.: Dependency-based evaluation of minipar. In Workshop on the Evaluation of Parsing Systems, Granada, Spain (1998)
41. Iftene, A., Balahur-Dobrescu, A.: Improving a QA System for Romanian Using Textual Entailment. In Proceedings of RANLP workshop "A Common Natural Language Processing Paradigm For Balkan Languages". ISBN 978-954-91743-8-0, September 26, Borovets, Bulgaria, 7-14 (2007)
42. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Mass (1998)

Towards Representation of the Discourse Structure Leading to Consensus

Tatiana Zidrasco, Victoria Bobicev

Technical University of Moldova, Stefan cel Mare,168,
2004, Chisinau, Moldova
tzidrashco@yahoo.com, vika@rol.md

Abstract. Consensus is the desired result in many argumentative discourses such as negotiations, public debates, and goal-oriented forums. This paper presents a summary of the work dedicated to investigating of discourse structure in order to determine rhetorical structures that lead to consensus. In addition we investigated language patterns extracted from the data collection in order to discover which ones are indicators of the following agreement.

Keywords: discourse analysis, rhetorical relations, consensus building.

1 Introduction

Since computer and web technologies offer vast opportunities for public debates, collaborative discussions, negotiations etc., the issue of consensus building within discourse has become more substantial. In computational linguistics there have been numerous studies dedicated to discourse analysis, modelling and analysis of collaboration, negotiations and agreement process [5], [6], [7], [18].

Two important components of discourse studies are the representation of discourse structure and language. We investigated discourse structure in attempt to find out how it can reflect successful or unsuccessful discussion. In this particular study we thought of a discussion as successful if participants achieve agreement about a statement. Our aim was to determine if there exist structures of discourse that lead to consensus and structures that do not lead to consensus. We think definition of such type of structures could help for better understanding of position and intentions of participants during agreement process. We performed our study using web-discussions (Wikipedia Talk pages, English language), where participants had as their goal to agree upon editing policy of Wikipedia articles. To build the discussion structure we used Rhetorical Structure Theory (RST) relations. We then applied statistical analysis to our discussions annotated with 918 relations.

As mentioned above, another important component of discourse analysis is language or better say, those words and phrases used by the participants to directly indicate the structure of the argument to the other participants. Thus we next investigated how language reflects success or failure in our web-discussions [18].

2 Related Works

There have been a number of researches of modelling and analyzing negotiation and agreement process in computational linguistics. In [5] multiagent collaborative planning discourse is analyzed and an artificial language is formulated for modelling such discourse. Modelling is done using proposal/acceptance and proposal/rejection sequences. Propose-Evaluation-Modify framework for collaboration is proposed in [6]. Slightly different approach to the problem of modelling of agreement process is described in [7]. They model their participant's collaborative behaviour according to Balance-Propose-Dispose agreement process and they focus on how information is exchanged in order to arrive to a proposal and what constitutes a proposal and it acceptance or rejection. We proposed to build discourse structure using RST and basing on empirical analysis, to determine which types of discourse structures are leading to final consensus. In [18] the preliminary study investigates how language reflects success or failure of electronic negotiations. They seek text characteristics which can help in prediction of negotiations success or failure. Using NLP and ML techniques they show how language differs in successful and failed negotiations. Thus we have also analyzed the discussion language in order to identify language features that influence the result in our discussions.

3 Web-Discussions Annotated with Rhetorical Relations

We stopped at Wikipedia discussions for two reasons: 1) these are web-mediated discussions; 2) these are task-oriented discussion - the purpose of each discussion is to achieve agreement about the final version of Wikipedia article; since we aimed to define discourse structures that lead to consensus, we considered these discussions to be suitable for our study.

3.1 Rhetorical Relations

Rhetorical Structure Theory is descriptive theory of hierarchical structure in discourse that identifies functional relationships between discourse parts based on the intentions behind their production [8]. In this study we present "discourse part" as participants' statements. Since one statement may contain different types of information; we segmented the statements into segments corresponding to *speech acts*. According to the definition, *speech act* is a term that refers to the act of successful communicating an intended understanding to the listener. Each speech act within one user's statement has a separate speech function like asking question, explaining, etc. One speech act can be related to one or more other speech acts. So in this study, "discourse parts" equivalent to *speech acts* become the *elementary segments* for annotation.

Although the application of RST for different types of conversational analysis is not novel, there is no common agreement on the application policy. The set of applied rhetorical relations is dependent on the purpose of the discourse analysis. We adapted the original RST set of relations to create our own tag set that we called *argumentation specific* rhetorical relations tag set.

The kinds of argumentation specific relations we used include Consensus relations *Agreement/Disagreement*, for example in (1) segment B states the agreement with the previous discussion segment A:

(1) Agreement

A: There is no official language of the United States. The correct answer to the question “What is the official language of the United States?” is “none”.

B: I agree with Nunh-huh.

We also introduce Question relations that in our opinion are necessary to connect question-answer pairs and help to determine the question intention, like in example (2), where relation *Require yes/no* is used to clarify the question intention stated in segment B:

(2) Require yes/no

A: The previous poster was absolutely correct. It needs to be permanently changed ASAP.

B: You want us to lock the page?

Our collection of web–discussions contained 1,764 statements (participants’ comments), the total number of participants was 506 and we obtained 918 rhetorical relations connecting the statements. We had only one annotator who annotated our discussions with the help of the annotation tool. The tool allows diagramming of the discussion structure. The annotation was done in two steps. First, the annotation tool structured the discussion into separated statements stated by various participants of the discussion. Then, using the list of the rhetorical relations proposed by the tool, annotator connected participants’ statements. One of the issues that arose during the annotation process was the ambiguity problem, when for one statement’s context more than one rhetorical relation definition was possible to apply. In some cases, the relation Unknown was used, as it was difficult to apply any rhetorical relation definition. In the Table 1 below we present our tag set of 27 rhetorical relations.

Table 1. Rhetorical Relations tag set.

Affirmation	Require evidence	Solution
Negation	Require detail	Warning
Evidence	Require yes/no	Concession
Justification	Request to do	Summary
Elaboration	Suggestion	Unknown
Explanation	Apology	Response
Background	Accusation	Addition
Example	Gratitude	
Agreement	Ironic_comment	
Disagreement	Offence	

3.2 Defining Discourse Structure that Leads to Consensus

We based on a simple assumption that within consensus building process discourse structure is regarded as successful, when there is a tendency for agreement. To formulate the assumption, we modeled our discourse as an oriented graph with nodes representing statements and arcs representing rhetorical relations that hold between statements. We supposed that it is possible to define successful discourse structure with help of sequences of rhetorical structures that hold between statements of the discussion. Here, we call these structures *agreement oriented*. For example, we presumed that the discourse sub-graph structures *Require evidence/Evidence* and *Evidence/Agreement* could be regarded as successful structures. In addition, we supposed that in successful discussions such rhetorical structures as *Evidence/Agreement* will be more frequent than let's say *Evidence/Disagreement* or *Suggestion /Agreement*.

Such heuristics needs to be verified empirically. So the validity of our assumptions will be observed from the further analysis of the discussion structures and shown in analysis results.

4 Rhetorical Structures Analysis

To verify the assumptions, we analyzed our data with help of so called sequence-based analysis. We counted frequency of rhetorical relations bigrams for *Agreement* (*Disagreement*) pairs and calculated priori

$$P(r_2|r_1) = C(r_1, r_2)/C(r_1) \qquad \textbf{(1)}$$

and posterior

$$P(r_1|r_2) = C(r_1, r_2)/C(r_2) \qquad \textbf{(2)}$$

probabilities, where, $C(r)$ and $C(r_1,r_2)$ denote frequencies of a rhetorical relation r and relation bigram (r_1, r_2), respectively. These calculations enabled us to identify rhetorical relations that most frequently precede *Agreement* and *Disagreement*. The results are presented in Table 2 and Table 3. Order of relation r_1 in the tables is sorted by $P(r_1|r_2)$ = Agreement, the posterior probability of r_1 when r_2 = Agreement, because this probability can be regarded as a contribution of r_1 for building consensus.

From the tables 2 and 3 it can be seen that most frequent Agreement pairs had *Evidence* as the relation that was followed by *Agreement*. The most frequent Disagreement pairs had *Suggestion* as the relation that was followed by *Disagreement*.

Also the frequency of the pairs *Evidence/Disagreement* is higher than *Evidence/Agreement*.

Next we applied Evidence-based analysis to investigate the influence of contribution (on this stage it is *Evidence*) relation on final agreement. The contribution relation r_1 is a target relation for analyzing its influence on final consensus relation.

Table 2. Priori and Posteriori Probability for Most Frequent Agreement Pairs.

Relation r_1	**$P(r_2$=Agreement$\mid r_1)$**		**$P(r_1 \mid r_2$= Agreement)**	
Evidence	0.176	(12/68)	0.072	(12/166)
Suggestion	0.170	(19 / 112)	0.114	(19/ 166)
Disagreement	0.133	(22 / 166)	0.133	(22/ 166)
Agreement	0.120	(20 / 166)	0.120	(20/ 166)
Answer	0.138	(4 / 29)	0.024	(4 / 166)
Explanation	0.107	(18 / 169)	0.108	(18/ 166)
Req_evidence	0.082	(4 / 49)	0.024	(4 / 166)
Justification	0.021	(1 / 47)	0.006	(1 / 166)

Table 3. Priori and Posteriori Probability for Most Frequent Disagreement Pairs.

Relation r_1	**$P(r_2$=Disagreement$\mid r_1)$**		**$P(r_1 \mid r_2$=Disagreement)**	
Evidence	0.221	(15/ 68)	0.090	(1 / 166)
Suggestion	0.277	(31/ 112)	0.187	(31/ 166)
Disagreement	0.127	(21/ 166)	0.127	(21/ 166)
Agreement	0.024	(4/ 166)	0.024	(4 / 166)
Answer	0.034	(1/ 29)	0.006	(1 / 166)
Explanation	0.077	(13/ 169)	0.078	(13/ 166)
Req_evidence	0	(0/ 49)	0	(0 / 166)
Justification	0.064	(3/ 47)	0.018	(3 / 166)

The consensus relation r_2 corresponds to *Agreement* or *Disagreement*. We calculated the probability of the bigram (r_1, r_2) to see the probability that *Agreement* would come after the *Evidence*. We considered the following two possibilities: when r_2 is *Agreement* (*Disagreement*), while r_1 is *Evidence* and when r_2 is *Agreement* (*Disagreement*), while r_1 is any other rhetorical relation. We compared ratios of appearing of *Agreement* and *Disagreement* in evidenced and non-evidenced pairs. The results of the Evidence-based analysis indicated only partial validity of our assumption about *Evidence* being the first relation followed by *Agreement*. Both sequence-based and evidence-based types of analysis only partially confirmed our assumptions.

5 Language Features

We next made another assumption, that language used in discussions has an impact on consensus building. We decided to analyze word unigrams, bigrams and trigrams in different types of statements. In [18] they demonstrated that there were characteristic words for successful and unsuccessful negotiations called “indicative words”. We made an attempt to make similar analysis for our collection of discussions from Wikipedia annotated with rhetorical relations.

Our text collection consisted of 320 files of Wikipedia discussion pages, some of them quite long, some rather short. The longest had more than 100 elementary segments; some short ones had just an exchange of two statements. Total number of word tokens was 148,948 and number of word types was 11,545. As it has already been mentioned statements were considered elementary segments and were annotated with rhetorical relations. It should be added that not every segment was annotated; some statements were left without annotation.

In [18] analysis of negotiations were based on the final result: success or failure of the negotiation; thus all discussion was considered as successful or unsuccessful. In our dialogue there was no final result; we concentrated on each statement as one unit with its rhetorical relation. Firstly, we made frequency dictionaries of words, word bigrams and word trigrams for all statements annotated with the same rhetorical relations. Quick analysis of these dictionaries revealed "indicative words" for the relations. For example, Disagreement is indicated with the higher rate of negations "not", "I don't", "there is no", "it is not", etc. Agreement on the contrary, had clear indicators: "I agree with", "have to agree". However, not all relation could be detected so easily; for example, Justification, Explanation, Suggestion had less specific words and much more content words referring to the discussed topic. As "indicative words" for these relations could be mentioned:

Justification – adverbs "reasonably", "rather", "as well";
Explanation – verbs "want to", "could be", "I feel";
Suggestion – "I think", "should be", "we should".

We selected all relations pairs r_1, r_2, where r_2 is Agreement or Disagreement and made frequency dictionaries for the texts of first relations which preceded Agreement or Disagreement respectively. Thereby we formed files with all words for the statements which were marked as, for example, Suggestion and preceded statements marked as Agreement. We created unigram, bigram and trigram frequency dictionaries for these statements. The next step was comparison of words for one type of statements which preceded Agreement and Disagreement respectively in order to reveal which words are indicative for the following agreement. In Table 4 some of the most frequent pairs of relations are presented, their indicative words and some comments are added.

In general, we observed that bigrams and trigrams of words which are indicative for agreement do not depend on relation. For all relations we investigated, specific features for Agreement are gentle, polite phrases. Also, to our surprise, pronouns have the great impact on following agreement: "we" is good indicator of agreement, while "you" indicate opposition, especially in phrases "you have" and "you should". We did not find verbs to be indicative words. Adverbs also have less impact on the result.

Table 4. Some of Frequent Pairs of Rhetorical Relations, their Indicative Words and Comments.

Relation bigram		Indicative words	Comments
r_1	r_2		
Suggestion	Disagreement	highly, quite, rather, reason is quite, should be, would be, better to	suggestions are more categorical and are formulated as from superior to inferior which provoke negation
Evidence	Agreement	we, if, a few, a certain, for the purposes, deem that, can cite some authority	less indicative words, more text about the topic, the language is more concrete and more gentle
Evidence	Disagreement	you due to, you need a, you will need, you'd have to	an aggressive language with many combinations of "you have", "you should", etc.

6 Conclusion

In the paper we present some results on the analysis of the relationship between rhetorical structure of the discourse and consensus building. We aimed to find structures of argumentative discourse that lead to agreement. We analysed a collection of web–discussions containing 1,764 statements with the total number of 506 participants and 918 rhetorical relations connecting the statements. We applied two types of statistical analysis sequence-based and Evidence-based. The results showed only partial consistency with our initial assumptions.

We also made an investigation of language used in discussions and its influence on the discussion outcome. The investigation of word unigrams, bigrams and trigrams showed that specific features of language which led to *Agreement* or *Disagreement* were similar indifferent which type of rhetorical relation preceded *Agreement* or *Disagreement* respectively. Actually, investigation of discourse structure and language for different types of relations should be a more extensive study. One of the most natural extensions of the study of language in discussion is more sophisticated statistical method application but our collection of discussions is comparatively small and data is rather sparse. Thus, we leave this study for the future when we obtain more annotated data. It is also good to mention that the results of such a study could be used for consensus facilitating function design in an argumentation support system.

References

1. Macintosh, A., Gordon, T. F., Renton, A.: Providing Argument Support for E-Participation. J. of Information Technology & Politics 6(1), 43-59 (2009)
2. Scheuer, O., Loll, F., Pinkwart, N., McLaren, B. M.: Computer-supported argumentation: A review of the state of the art. International Society of the Learning Sciences Inc, Int. J. of Computer-Supported Collaborative Learning (IJCSCL) 5(1), 43-102 (2010)
3. Verheij, B.: Artificial Argument Assistants for Defeasible Argumentation. Artificial Intelligence 150 (1-2), 291-324 (2001)
4. Reed, C., Rowe, G.: Araucaria: Software for Argument Analysis, Diagramming and Representation. Int. J. of AI Tools 14, 961-980 (2004)
5. Sidner, C. L.: An Artificial Discourse Language for Collaborative Negotiation. In: Proc. 12th Nat. Conf. on Artificial Intelligence, 814-819 (1994)
6. Chu-Carroll, J., Carberry, S.: Collaborative Response Generation in Planning Dialogues. Computational Linguistics 24(3), 355-400 (1998)
7. Di Eugenio, B., Thomason, R. H., Jordan, P. W., Moore, J. D.: The Agreement Process: an Empirical Investigation of Human-Human Computer-Mediated Collaborative Dialogues. Int. J. of Human Computer Studies 53(6), 1017-1076 (2000)
8. Stent, A.: Rhetorical Structure in Dialog, Proc. 1st Int. Conf. on Natural Language Generation (INLG'2000), Mitzpe Ramon, Israel, 247-252 (2000)
9. Daradoumis, T.: Towards a Representation of the Rhetorical structure of Interrupted Exchanges. Trends in Natural Language Generation: An Artificial Intelligence Perspective, Springer, Berlin, 106-124 (1996)
10. Hashida, K.: Semantic Authoring and Semantic Computing. In: A. Sakurai, K. Hashida, K. Nitta (eds.) Artificial Intelligence: Joint Proc. of the 17th and 18th An. Conf. of the Japanese Society for Artificial Intelligence. LNCS vol. 3609, Springer, 137-149 (2007)
11. Mann, W. C., Thompson, S. A.: Rhetorical Structure Theory: A Theory of Text Organization, Reprinted from the Structure of Discourse. ISI: Information Sciences Institute, Los Angeles, CA, ISI/RS-87-190, Reprinted from The Structure of Discourse, 1-81, (1987)
12. Carlson, L., Marcu, D., Okurowski, M. E.: Building a Discourse-Tagged Corpus in the Framework o Rhetorical Structure Theory. In: Current Directions in Discourse and Dialogue, J. van Kuppevelt and R. Smith (eds.). Kluwer Academic Publishers, 85-112 (2003)
13. Jovanovic, N., Rieks op den Akker, Nijholt, A.: A Corpus for Studying Addressing Behaviour in Multi-Party Dialogues. Springer Science + Business Media B.V. (2006)
14. Azar, M.: Argumentative Text as Rhetorical Structure: An Application of Rhetorical Structure Theory. Argumentation 13(1), 97-144 (1999)
15. Moore J. D., Paris C. L.: Planning text for advisory dialogues: Capturing Intentional and Rhetorical Information. In: Computational linguistics - Association for Computational Linguistics, MIT Press, MA, ETATS-UNIS, Cambridge, 651-694 (1993)
16. Taboada, M., Mann, W. C.: Applications of Rhetorical Structure Theory. Discourse Studies 8 (4), 567-588 (2005)
17. Tokuhisa, R., Terashima, R.: Relationship between Utterances and "Enthusiasm" in Non-task-oriented Conversational Dialogue. In: Proc. 7th SIGDIAL Workshop on Discourse and Dialogue, Sydney, 161-167 (2006)
18. Sokolova, M., Szpakowicz, S., Nastase, V.: Using Language to Determine Success in Negotiations: A Preliminary Study. Proceedings of the 17th Canadian Conference on Artificial Intelligence (Canadian AI'2004), Springer, 449-453 (2004)
19. Alexy, R.: Problems of Discourse Theory. Critica, Revista Hispanoamericana de Filosofia. vol. XX, No58, 43-65 (1988)

Understanding the Web using Natural Language Semantics

Diana Trandabăţ[1, 2]

[1] Faculty of Computer Science, University “Al. I. Cuza” of Iași
[2] Institute of Computer Science, Romanian Academy, Iasi Branch
dtrandabat@info.uaic.ro

Abstract. This paper presents a semantic role labeling system acting as the backbone of an application that establishes the roles that entities have in different contexts, and what are the temporal, modal or local constraint that determine or restrict an event to take place. A semantic role represents the relationship between a predicate and an argument. Semantic parsing, by identifying and classifying the semantic entities in context and the relations between them, has great potential on its downstream applications, such as text summarization or machine translation.

Keywords: semantic role labelling, semantic labelling classifications, text mining, knowledge extraction.

1 Introduction

In this paper we propose a system which, starting from an input entity, extracts web pages found on a Google search of the particular entity, selects the snippets that contain the input entity, and then performs semantic role labeling to extract the relations between the entity and its context.

Thus, our system created a contextual map by identifying the role an entity plays in different contexts, as well as the roles played by words frequently co-occurring with the input entity.

The motivation behind the work presented in this paper is the need to create a map of structured context related to a specific entity (e.g. a company or product name, an event, etc.). Through this map, the concepts that are usually in relation to the searched input entity are highlighted, together with their specific role (which can be of type Cause, Effect, Location, Time, etc.), thus providing a good material for social analyses, market research or other marketing purposes.

The paper is structured in 7 sections. After an introduction in the field of semantic role analysis, we briefly present the current work in Section 2. Section 3 presents a classification of semantic role labeling programs, while Section 4 introduces the overall application, describing the intermediary steps. Section 5 presents our approach for a Semantic Role Labeling system, which is evaluated in Section 6. The final section draws the conclusions of this paper and discusses further envisaged developments.

2 Semantic Roles

Fillmore in [7] defined six semantic roles: *Agent*, *Instrument*, *Dative*, *Factive*, *Object* and *Location*, also called *deep cases*. His later work on lexical semantics led to the conviction that a small fixed set of deep case roles was not sufficient to characterize the complementation properties of lexical items, therefore he added *Experiencer*, *Comitative*, *Location*, *Path*, *Source*, *Goal* and *Temporal*, and then other cases. This ultimately led to the theory of Frame Semantics [6], which later evolved into the FrameNet project [1]. In the last decades, hand-tagged corpora that encode such information for the English language were developed (VerbNet [13], FrameNet and PropBank [18]). For other languages, such as German, Spanish, and Japanese, semantic roles resources are being developed. For Romanian, [22] has started to automatically build such a resource.

For role semantics to become relevant for language technology, robust and accurate methods for automatic semantic role assignment are needed. With the SensEval-3 competition - http://www.senseval.org/ - and the CONLL Shared Tasks - http://ifarm.nl/signll/conll, Automatic Labeling of Semantic Roles, identifying frame elements within a sentence and tag them with appropriate semantic roles given a sentence, a target word and its frame [14], has become increasingly present among researchers worldwide. Most general formulation of the Semantic Role Labeling (SRL) problem supposed determining a labeling on (usually but not always contiguous) substrings (phrases) of the sentence *s*, given a predicate *p*, as in the following example:

In recent years, a number of studies, such as [3] and [8], have investigated this task on the FrameNet corpus. Role assignment has generally been modeled as a classification task: A statistical model is trained on manually annotated data and later assigns a role label out of a fixed set to every constituent in new, unlabelled sentences. The work on SRL has included a broad spectrum of probabilistic and machine-learning approaches to the task, from probability estimation [8], through decision trees [21] and support vector machines [20], to memory-based learning [15]. While using different statistical frameworks, most studies have largely converged on a common set of features to base their decisions on, namely syntactic information (path from predicate to constituent, phrasal type of constituent) and lexical information (head word of the constituent, predicate).

Early SRL models attempted to predict argument roles in new data, looking for the highest probability assignment of roles r_i to all constituents i in the sentence, given the set

$$F_i = \{phrase\_type_i, path_i, governing\_category_i, position_i,$$
$$voice_i, head\_word_i\}$$

of features for each constituent in the parse tree, and the predicate *p*:

$$\arg\max_{r1..n} P(r_{1..n}|F_{1..n}p) \qquad \textbf{(1)}$$

The probability estimation is broken into two parts, the first being the probability *$P(r_i|F_i,p)$* of a constituent's role given our five features for the constituent, and the predicate p. Due to the sparsity of the data, probabilities are estimated from various subsets of the features, and interpolated as a linear combination of the resulting distributions. The interpolation is performed over the most specific distributions for which data are available, which can be thought of as choosing the topmost distributions available from a backoff lattice.

The probabilities $P(r_i \mid F_{1..n}, p)$ are combined with the probabilities $P(\{r_i \mid F_{1..n}, p\})$ for a set of roles appearing in a sentence given a predicate, using the following formula:

$$P(r_{1..n} \mid F_{1..n}, p) \approx P(\{r_{1..n}\} \mid p)\prod_i \frac{P(r_i \mid F_i, p)}{P(r_i \mid p)} \qquad \textbf{(2)}$$

In other research, SVMs performed well on text classification tasks [20], which is probably why many researchers have used them for semantic role classification. Training instances in SVM classification are represented as sparse feature vectors in a high dimensional space. Suppose training instances belong either to positive or negative class as follows:

$$\begin{aligned}&(\vec{x}_1, y_1),...,(\vec{x}_n, y_n)\\ &x_i \in \Re^n, y_i \in \{+1,-1\}\end{aligned} \qquad \textbf{(3)}$$

$\vec{x}_i$ is a *n* dimensional feature vector of the *i*-th sample: $\vec{x}_i = (f_1, \ldots, f_n) \in \Re^n$, y_i is a scalar value that specifies the class (positive (+1) or negative (-1)) of *i*-th data. Classification can be described as building the function f : $\Re^n \rightarrow \{\pm 1\}$ by going through a learning process, under the assumption that new examples were generated from the same unknown probability distribution P($\vec{x}$, *y*) [12].

A SVM classifier tries to separate positive from negative examples by finding a hyperplane

$$(\vec{w} \cdot \vec{x}) + b = 0, w, x \in \Re^n, b \in \Re \qquad \textbf{(4)}$$

that divides the training data in two groups. SVMs try to optimize parameters *w* and *b* to find the optimal solution. Optimal means finding the hyperplane separating training examples with the maximal margin. Margin can be defined informally by the distance between the hyperplane and the boundaries in which it can move without any misclassification.

A problem with using SVMs for the SRL task is that SVMs are binary classifiers, i.e. they can only differentiate between two classes. Two approaches have been developed to overcome this problem in SRL [20]:

- Pairwise approach - A separate binary classifier is trained for each of the class pairs and their outputs are combined to predict the classes. The total number of classifiers required for this approach is $N*(N-1)/2$.
- One versus all (OVA) approach - *n* classifiers are trained for a *n*-class problem. Each classifier can discriminate between a particular class and the set of all other classes. Another problem is that kernel-based systems such as SVMs, exhibit classification speeds which are substantially lower than those of other machine learning algorithms [12].

A commonly adapted approach to reduce training time, is to filter out the instances that have a very high probability of being null.

Other approaches used memory based learning to reason on the basis of similarity of new situations to earlier encountered situations [4]. The learning component of a MBL system is memory based: all training examples are stored in memory. The other component of a MBL system, the performance component, is similarity-based and performs the actual classification.

During training, training instances are loaded into memory. An instance consists of a vector containing feature-value pairs and a class assignment. During classification, unseen examples are compared to instances in the training data. This comparison is done using a distance metric $\Delta(X,Y)$. The class assignment is based on the k-nearest neighbours algorithm: the most common class amongst the k most similar training instances is chosen. In case of a tie among categories, a tie breaking resolution method is used.

Different distance metrics can be used in MBL. The most common of which is the overlap metric:

$$\Delta(X,Y)=\sum_{i=0}^{n}\delta(x_i,y_i) \qquad \textbf{(5)}$$

$$\delta(x_i,y_i)=\begin{cases} abs(\frac{x_i-y_i}{\max_i-\min_i}) & \text{if numeric, else} \\ 0 & \text{if } x_i=\mathrm{y_i} \\ 1 & \text{if } x_i\neq \mathrm{y_i} \end{cases} \qquad \textbf{(6)}$$

A metric k-NN algorithm is called IB1. In the overlap metric, all features have the same weight. To improve performance, domain knowledge can be used to assign different weights to different features. Weights can also be determined by calculating statistics (e.g. frequencies) of features in the training data. These statistics can be used to determine which features are good predictors of the class labels and which features are less relevant. A useful tool for measuring feature relevance that is especially beneficial for NLP tasks, is information gain (IG) [4].

Information Gain looks at each feature and measures how much information it contributes to the knowledge needed to predict the correct class label. The most common way of measuring the IG of a feature *i* is to compute the difference in uncertainty (i.e. entropy) between situations without and with knowledge of the value of that feature:

$$w_i = H(C) - \sum_{v \in V_i} P(v) \times H(C \mid v) \qquad \textbf{(7)}$$

where *C* is the set of class labels, V_i is the set of values for feature *i*, and *H(C)* is the entropy of the class labels. Entropy measures the amount of uncertainty of a variable, and is defined as

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \qquad \textbf{(8)}$$

The probabilities are estimated from relative frequencies in the training set. The computation of IG in the definition above is symbol based, for real values an intermediate step must be taken.

An important drawback of the presented systems is that they don't treat nominal predicates, being only built for verbal predicates. Furthermore, they only consider one predicate per sentence, even if this is not always the case. For example, in the sentence `The awarding of the Nobel Prize to President Obama was largely debated`, we have two predicate words, `the awarding`, having as *Theme* `of the Nobel Prize`, and `debated`, having two arguments, an *Agent* `The awarding of the Nobel Prize to President Obama` and an *Manner* complement, `largely`.

The semantic role labeling system presented in this paper considers both verbal and nominal predicates (see more details in Section 5).

3 Semantic Role Labelers Classifications

The different approaches to semantic role labelers found in the literature can be divided into four categories, with respect to the type of tokens they classify:

- constituent-by-constituent (C-by-C);
- phrase-by-phrase (P-by-P);
- word-by-word (W-by-W);
- relation-by-relation (R-by-R).

In the **C-by-C semantic role labeling**, the syntactic tree representation of a sentence is linearized into a sequence of its syntactic constituents (non-terminals). Then each constituent is classified into one of several semantic roles using a number of features derived from the sentence structure or a linguistic context defined for the constituent token.

In the **P-by-P and W-by-W methods**, described in [9], the problem is formulated as a chunking task and the features are derived for each base phrase and word, respectively. The tokens are classified into one of the semantic labels using an IOB (inside-outside-begin) representation and a set of classifiers, one for each class.

The **R-by-R method** is based on dependency trees generated from constituency trees. Although these systems do not use more information than C-by-C systems, the information is structured in a different manner and, consequently, the nature of some linguistic features is quite different. Hacioglu points out in [10] that this restructured information is very useful in localizing the semantic roles associated with the selected predicate, since the dependency trees directly encode the argument structure of lexical units populated at their nodes through dependency relations.

Types of vector spaces for semantic role labeling

When applying machine learning techniques to semantic roles, one assumes that frame semantic resource can be modeled and represented by a suitable semantic vector space model. One approach, proposed in [19], is to represent semantic frames and semantic roles as distributional vectors in the space - i.e. by using co-occurrence vectors.

Vector space models (VSM) are widely used in NLP for representing the meaning of words or other lexical entities. The basic intuition is that the meaning of a word is defined by the context in which it appears. The context can be defined in different ways: as the set of words around the target word, as the paragraph in which it appears, the document, and so on. Vector spaces are used to model this intuition, by collecting statistics about the contexts of a target word within a large corpus.

Following this idea, a target word *tw* is represented by a vector, whose dimensions are the contexts in which it appears. When contexts are words *w* in an *n*-window of the target - i.e. *tw* co-occurs with the context word if *w* is within *n*-tokens on the left or on the right - we have a **word-based** VSM. When contexts are documents or sentences in which the target appears, we talk about **document-based VSM**. The value of each dimension is given by the co- occurrence value of the target word with the given context. In the simplest case, these values are counts, i.e. the number of times that the target and the context co-occur in the corpus. More sophisticated association measures can be used to compute co-occurrence values. The most widely used are conditional probability of the $p(w \mid tw)$ word given the target or point-wise mutual information $pmi(tw, w)$ between the target and the word.

In our particular setting, we focus on word-based and syntax-based spaces, using constituent-by-constituent semantic role labeling.

4 Towards Semantically Interpreting Texts

The goal of the application we propose is to extract the context in which an entity occurs in web documents, together with the relations that the searched entity establishes with frequently co-occurring words. The basic architecture of our application contains a series of specialized modules, as follow:

User input acquiring module

This module prompt the user for an input entity, usually representing a company or product name, an event name, a person, etc. A drop down list of the most frequent searched entities is offered as suggestion.

Web page retrieval module

This module extract from the web the first n (in our tests *n*=200) web pages found on a Google search for the input entity. Google search options restricting the web search can be applied, such as selecting articles from newspapers, blogs or in a specific language.

Snippet extraction module

Using the Google snippet suggestion and some simple heuristics, the paragraphs containing the input entity are selected. A simple anaphora resolution method, based on a set of reference rules, is applied to the web pages, in order to link all entities to their referees.

The anaphoric system we used is a basic rule-based one, focusing on named entity anaphoric relations. Thus, we developed a rule-based system that performs the following actions:

- identifies a subset of a named entity with the full named entity, if it appears as such in the same text. For instance, *Caesar* is identified with *Julius Caesar* if both entities appear in the same text. Similarly, *the President of Romania* and *the President* are considered anaphoric relations of the same entity, if they appear in a narrow word window in the text.
- solves acronyms using a gazetteer we have initially built over the Internet, and which is continuously growing in size. For instance, *United States of America* and *USA* are co-references.
- searches for different addressing modalities and matches the ones that are similar. For instance, *John Smith* is co-referenced with *Mr. Smith*, and *Mary and John Smith* is co-referenced with *The Smiths*, or *The Smith Family*.
- solve pronominal anaphora in a simplistic way. Thus, if a pronoun (i.e. *she*, *he*, *him*, *his* etc.) is found in the text, and in the preceding sentence an entity is found, then we create an anaphoric link between the pronoun and its antecedent. A similar rule exists for companies, where the pronoun *it* may be linked to *the Insurance Company*, for instance.

Snippet cleaning module

After relevant paragraphs (containing the input word) are extracted, functional words such as (and, so, a, etc.) are eliminated from these paragraphs, since, being statistically too frequent for any kind of texts, they do not convey any useful information for semantic role labeling. A list of these functional words, created by using word frequencies in large corpora, is used.

Semantic role labeling module

This module performs semantic role labeling on the obtained paragraphs, in order to identify the role the entity in question and the related entities plays. It will be described in details in the following section and evaluated in Section 6.

Module for the creation of a map of concepts

This modules works in two steps: first, it extracts from the semantic role analysis the relations between the searched entity and the neighbour entities, creating a list of relations. Secondly, it generalizes the concepts that are found to be in relation with the searched entity across all extracted paragraphs, using the WordNet [5] hierarchy.

The next section presents the core module of this architecture, the semantic role labeling system, developed by training a set of supervised machine learning algorithms for several languages.

5 Our Semantic Role Labeling Approach

In order to detach semantic information from texts, we considered building a supervised SRL system. Several machine learning techniques and feature sets have been tested using the algorithms implemented in the Weka toolkit [25]. We used 12 of the most common machine learning algorithm that are available in Weka, such as decision trees, SVMs, memory-based learners, etc. A full description of the machine learning algorithm from Weka used for PASRL can be found in [24].

The final system developed into a “platform" for creating supervised Semantic Role Labeling systems using an annotated training corpus. The platform (which we named PASRL – Platform for Adjustable Semantic Role Labeling) trains the 12 classifiers with different feature sets, checks the performances of the different obtained models and selects the best performing model.

The 10 fold cross-validation results of all classifiers are also saved since they provide a confusion matrix that can be used to see which classes were correctly predicted by different classifiers. The output of PASRL is a Semantic Role Labeling System, a sequence of trained models which can be used to annotate new texts.

PASRL was developed using training sets for different languages: English, German, Chinese, Czech and Japanese, provided for research purposes by the CoNLL 2009 shared task. The training data consisting of manually annotated treebanks such as the Penn Treebank for English, the Prague Dependency Treebank for Czech and similar treebanks for Chinese, German and Japanese languages, enriched with semantic relations.

The training data contains syntactic and dependency information, detailed in Table 1. PASRL is composed of two main sub-systems: a Predicate Prediction module and an Argument Prediction module.

Table 1. Description of the Input Format for PASRL.

No.	Name	Description
1	ID	Token counter, starting at 1 for each new sentence.
2	FORM	Word form or punctuation symbol.
3	LEMMA	Lemma or stem (depending on particular data set) of word form, or an underscore, if not available.
4	POS	Part-of-speech tag, where the tagset depends on the language, or identical to the coarse-grained part-of-speech tag, if not available.
5	HEAD	Head of the current token, which is either a value of ID (meaning that the word with the ID n is the head-word of the current token) or zero ("0", if the head word is the ROOT fictive node).
6	DEPREL	Dependency relation to the HEAD. The set of dependency relations depends on the particular language.

Predicate Prediction module

The first module of the semantic role labeling system is the *predicate prediction* module. The Predicate Prediction module system is composed out of three sub-modules:

- *Predicate Identification* – this module takes the syntactic analyzed sentence and decides which of its verbs and nouns are predicational (can be predicates), thus for which ones semantic roles need to be identified;
- *Predicate Sense Identification* – once the predicates for a sentence are marked, each predicate need to be disambiguated since, for a given predicate, different sense may demand different types of semantic roles;
- *Joint Predicate and Predicate Sense Identification* – jointly identifies the predicates and their senses (the two above sub-modules).

Our semantic role labeling system uses the PropBank [18] annotation of semantic roles. Since predicational words are not just verbs, beside PropBank for the verbal frames, NomBank [20] is also used for nouns. For example, the verb to be has no annotation in PropBank, since it is a state and not an action, predicational, verb. Similarly, the NomBank is used to sort nouns that can behave as predicates from those that cannot have semantic arguments.

The output of this module is the input file, where each verb or noun that behaves as a predicate is annotated. After the predicates from the input sentence are identified, the next module is successively applied for all the predicates found in the sentence, in order to identify for all of them all and only their arguments.

The PASRL platform is described in more details in [23] and [24].

Argument Prediction module

After running the first module of the semantic role labeling system (either pipelined or joint), the *Argument Identification* task is called in order to assign to each syntactic constituent of a verb / noun its corresponding semantic role. The instances in this case are not only the nouns and verbs, but every word in the sentence.

The Argument Prediction module performs argument prediction, based on the dependency relations previously annotated with the MaltParser [16] and the Predicate Prediction output. The input of this module contains syntactic information (part of speech and syntactic dependencies), predicate and predicate role set, and in the output each syntactic dependent of the verb is labeled with its corresponding role. This module uses as external resources PropBank and NomBank frame files and the WordNet ontology [5] to extract for each word its hypernym.

For each module, the set of 12 classifiers from Weka framework are trained. After running all the classifiers for all the modules, their performance is compared, and the module that obtains the highest summed performance is considered the best. The models for this best configuration are saved, and the best path is written to a configuration file. This configuration can then be used at a later time to annotate new texts with the developed SRL system. If all the created models are saved, and not just the best performing ones, the user can define, using the configuration file, the sequence of classifiers he wishes to run for each subtask in order to annotate new texts using the pre-trained models

6 Evaluating PASRL

An evaluation metric for semantic roles have been proposed within CoNLL shared task on semantic role labeling [21]. The semantic frames are evaluated by reducing them to semantic dependencies from the predicate to all its individual arguments. These dependencies are labeled with the labels of the corresponding arguments. Additionally, a semantic dependency from each predicate to a virtual ROOT node is created. The latter dependencies are labeled with the predicate senses.

The evaluation of the PASRL performance was computed using 10-fold cross-validation on the training set. For each task, PASRL evaluates all the machine learning algorithms used against the gold-annotated corpus, and the best performing algorithm is saved in a configuration file. The evaluation was performed considering the number of correctly classified labels and correctly identified predicates. PASRL was tested using the training data for different languages, available through the ConLL 2009 Shared Task: English, German, Czech, Chinese and Japanese.

The overall results showed that the *Argument Prediction* was easier than the *Predicate Sense Identification* task, since better results were obtained for most of the algorithms in the first problem. Detailed results for the *Predicate Prediction* task and its sub-tasks are presented below.

For the *Predicate Identification* task, running the classifiers with the default weights of Weka for the English dataset, their results ranged from 86% correctly identified predicates to 56%. The algorithm that performs best is the J48 classifier (decision tree) and the one that performs worst is the simple ZeroR classifier (see Table 2).

Table 2. Top 5 ML algorithms evaluated with 10-fold cross-validation for English, for the Predicate Identification task.

ML Algorithm	10-fold cross validation
J48	86.323
AdaBoostM1	86.016
AttributeSelectedClass	82.169
DecisionTable	80.921
KStar	78.97

Using boosting techniques with J48 as base classifier could improve further the module's performance. Changing the default weights of the classifiers can modify their performances, but we believe that the hierarchy will not change substantially. However, this remains a direction to address in a further work.

The best performing model (J48 in this case, which is a decision tree learning method) is saved and will be used when the *Predicate Identification* module will be called from the configuration file for annotating an unlabeled text.

Table 3. Top 5 ML algorithms for the Predicate Sense Identification task for English.

ML Algorithm	10-fold cross validation
LogitBoost	61.085
J48	60.641
HyperPipes	58.868
AdaBoostM1	58.322
DecisionTable	57.667

The results for the *Predicate Sense Identification* Task are considerably worse than the ones for *Predicate Identification* task (see Table 3), even if the results report an evaluation performed on a gold annotated input file. Therefore, the actual results are expected to be even worse. However, we notice that J48 is still among the best algorithms, and memory based algorithms generally perform badly on this subtask.

Table 4. Top 5 ML algorithms for the Joint Predicate Identification and Predicate Sense Identification task for English.

ML Algorithm	10-fold cross validation
J48	57.974
AttributeSelectedClass	57.899
DecisionTable	57.667
AdaBoostM1	56.548
IBk	51.200

Instead of running the *Predicate Identification* and the *Predicate Sense Identification* processes successively, we tested running them simultaneously, using the same features presented above. The results (Table 4) show an even worst performance than for *Predicate Sense Identification*, suggesting that the option of running successively the *Predicate Identification*, followed by the *Predicate Sense Identification* modules may be better than running the joint task (barely 58% for the best).

After PASRL is trained, it can be used to annotate raw text with the models saved. A configuration file is used to tell the program which models will be run. The best models created during the development phase of PASRL are offered as pre-trained models, if the user only wants to use PASRL as a semantic role labeler, and not a semantic role labeling model creator. The plain text is preprocessed in order to add syntactic and dependency information. For English, the sequence is: Stanford Parser [7] for Part-of-speech identification and MaltParser for dependency relations. For languages other than English, the MaltParser can be trained on the training corpus provided by the CoNLL 2009 Shared Task organizers. However, Stanford Parser has build-in language models for English, German and Chinese, for the other languages, a language-specific POS-tagger is needed.

When evaluating the pre-trained models for English on new data, using the whole processing chain (including part of speech and dependency annotation), the results are promising, with 68% for noun predicate and 81% for verb predicate F1.

7 Conclusions

This paper presented a semantic role labeling system developed using supervised machine learning algorithms from the Weka framework. This system is used in an application that monitors the contexts in which a specific entity appears in web texts and the relations it has with other co-occurring concepts. The developed SRL platform can be used for different languages, provided that a training corpus annotated with semantic roles is available. After testing several classifiers on different subproblems of the SRL task (*Predicate Identification*, *Predicate Sense Identification, Predicate and Sense Identification*, *Argument Prediction*), the proposed system chooses the algorithm with the greatest performance and returns a Semantic Role Labeling System (a sequence of trained models to run on new data).

The envisaged application of our system is in the marketing field, where the related concepts our system offers can be used to monitor the reaction of the consumer to different changes in a company's marketing policies.

Acknowledgments. The research presented in this paper is funded by the Sectoral Operational Program for Human Resources Development through the project "Development of the innovation capacity and increasing of the research impact through post-doctoral programs" POSDRU/89/1.5/S/49944 and also by a grant of the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-RU-PD-2011-3-0292, contract no. 55/28.11.2011.

References

1. Baker, G., Collin, F., Fillmore, C. J., Lowe, J. B.: The Berkeley FrameNet project. In Proceedings of COLING-ACL, Montreal, Canada (1998)
2. Budanitsky, A., Graeme, H.: Evaluating wordnet-based measures of semantic distance. Computational Linguistics, 32(1):1347, (2006)
3. Chen, J., Rambow, O. Use of deep linguistic features for the recognition and labeling of semantic arguments. In Proceedings of EMNLP (2003)
4. Daelemans, W., Zavrel, J., van der Sloot, K., van den Bosch, A.: TiMBL: Tilburg Memory Based Learner, version 5.1, Reference Guide. Technical report, ILK Technical Report Series 04-02, (2003)
5. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press (1998)
6. Fillmore, C. J.: Frame semantics. In Linguistics in the Morning Calm, Hanshin Publishing, Seoul, 111-137 (1982)
7. Fillmore, C. J.: The case for case. In Bach and Harms, editors, Universals in Linguistic Theory, Holt, Rinehart, and Winston, New York, 1-88, (1968)
8. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Computational Linguistics, 28(3), 245-288 (2002)
9. Hacioglu, K., Ward, W.: Target word detection and semantic role chunking using support vector machines. In Proc. of HLT/NAACL-03 (2003)
10. Hacioglu, K.: Semantic role labeling using dependency trees. In Proceedings of the 20th international conference on Computational Linguistics COLING'04, Morristown, NJ, USA (2004)
11. Klein, D., Manning, C. D.: Accurate unlexicalized parsing. In Proceedings of the 41st Meeting of the Association for Computational Linguistics, 423-430 (2003)
12. Kudo, T. Machine Learning and Data Mining Approaches to Practical Natural Language Processing. PhD thesis, School of Information Science, Nara Institute of Science and Technology (2003)
13. Levin, B. Hovav, M. R.: Argument Realization. Research Surveys in Linguistics Series. Cambridge University Press, Cambridge, UK (2005)
14. Marquez, L., Xavier, C., Litkowski, K. C., Stevenson, S. Semantic role labeling: An introduction to the Special Issue. Computational Linguistics, 34(2), 145-159 (2008)
15. Morante, R., Daelemans, W., Van Asch, V.: A combined memory-based semantic role labeler of English. In Proceedings of CoNLL, Manchester, UK, 208-212 (2008)
16. Nivre, J.: An efficient algorithm for projective dependency parsing. In Proc. of the 8th International Workshop on Parsing Technologies (IWPT 03), 149-160 (2003)
17. Pado, S. Lapata, M.: Dependency-based construction of semantic space models. Computational Linguistics, 33(2) (2007)
18. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: An annotated corpus of semantic roles. Computational Linguistics, 31(1), 71-106 (2005)
19. Pennacchiotti, M., De Cao, D., Marocco, P., Basili, R.: Towards a Vector Space Model for FrameNet-like Resources. In Proceedings of LREC'08, Marrakech, Morocco (2008)
20. Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J. H., Jurafsky, D.: Support vector learning for semantic argument classification. Machine Learning Journal, 60(13), 11-39 (2005)
21. Surdeanu, M., Harabagiu, S., Williams, J., Aarseth, P.: Using predicate-argument structures for information extraction. In Proceedings of ACL2003, Tokyo, 8-15 (2003)

22. Trandabăț, D., Husarciuc, M.: Romanian semantic role resource. In Proceedings LREC'08, Marrakech, Morocco, May (2008)
23. Trandabăț, D.: Extracting Semantic Information from Texts, in Proceedings of the 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC2011, Timisoara, Romania (2011)
24. Trandabăț, D.: Natural language processing using semantic frames. PhD Thesis, 2010, http://students.info.uaic.ro/~dtrandabat/thesis.pdf (2010)
25. Witten, I. H., Eibe, F.: Data Mining: Practical Machine Learning Tools and Techniques (Second Edition). Morgan Kaufmann (2005)

INDEX OF AUTHORS

Bobicev Victoria 61

Colhon Mihaela 11

Cristea Dan 39

Gîfu Daniela 39

Gîscă Veronica 29

Iftene Adrian 49

Palade Olga 29

Petic Mircea 29

Simionescu Radu 21

Trandabăț Diana 69

Zidrasco Tatiana 61

PROGRAMME OF THE WORKSHOP

Language Resources and Tools with Industrial Applications

Tuesday, 30.08.2011
Room 335

17.45-18.00

A Contrastive Study of Syntactic Constituents in English and Romanian Texts
Mihaela Colhon, University of Craiova
mghindeanu@inf.ucv.ro

18.00 – 18.15

Hybrid POS Tagger
Radu Simionescu, "Alexandru Ioan Cuza" University, Faculty of Computer Science
radu.simionescu@info.uaic.ro

18.15 – 18.30

Multilingual Mechanisms in Computational Derivational Morphology
Mircea Petic, Veronica Gisca and Olga Palade, Institute of Mathematics and Computer Science, Republic of Moldova Academy of Sciences
mirsha@pisem.net, veronica.gisca@gmail.com, palade.olga@gmail.com

18.30 – 18.45

Using Natural Language Technology to Measure Mass Media Reactions in the Election Context
Daniela Gifu and Dan Cristea, "Alexandru Ioan Cuza" University, Faculty of Computer Science
daniela.gifu@info.uaic.ro, dcristea@info.uaic.ro

18.45 – 19.00

Towards Representation of the Discourse Structure Leading to Consensus
Tatiana Zidrasco and Victoria Bobicev, Technical University of Moldova
tzidrashco@yahoo.com, vika@rol.md

Wednesday, 31.08.2011
Room 336

17.45-18.00

Using Textual Entailment in Internet Surveillance
Adrian Iftene, "Alexandru Ioan Cuza" University, Faculty of Computer Science
adiftene@info.uaic.ro

18.15 – 18.30

Understanding the Web using Natural Language Semantics
Diana Trandabăț, "Alexandru Ioan Cuza" University, Faculty of Computer Science
dtrandabat@info.uaic.ro

18.30– 19.30
The SENTIMATIX Project – round table on current state and further steps