“Al. I. Cuza” University of Iasi

Faculty of Computer Science

Predication Driven Textual Entailment

2011

PhD Student Mihai-Alex Moruz

Supervisor Prof. Dr. Dan Cristea

Thesis version updated and modified according to the recommendations of the reviewers.

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Dan Cristea for guiding my steps as a researcher, for helping me expand my horizons and for always pushing me to better myself. I would also like to thank him for the confidence he had in my abilities, which helped me through the highs and the lows of these years, and for the insights he provided with regards to my research.

Special thanks go out to my colleagues from the Natural Language Processing group within the faculty of Computer Science, with whom I have shared these past years of researching, attending conferences, talking and laughter, and who made these years such joy: Adi Iftene (to whom I would like to thank again for all the talks and collaboration), Maria Husarciuc (who is now Maria Moruz, my loving wife), Diana Trandabăţ, Ionuţ Pistol, Lucian Gâdioi, Iustin Dornescu and Marius Răschip. Thank you all for your help, for your advice and for all the things that I managed to learn from you all.

I am also very grateful to my colleagues at the Institute for Computer Science within the Romanian Academy, especially to Neculai Curteanu, who helped me become a better researcher by teaching me how to read and write scientific papers, how to be thorough and how follow through with my ideas. Thank you for the years of patience and the wonderful discussions.

Many thanks to prof. Dan Tufiş, for giving me such valuable advice throughout my PhD and for always taking the time to answer all of my questions, to prof. Bernardo Magnini, for helping me shape some of the fundamental ideas of my thesis, and to prof. Dorel Lucanu and prof. Cornelius Croitoru, for their insightful comments and advice.

I would also like to offer special thanks to my family, for their patience and understanding, for putting up with the long days and nights of work, and for the help they so graciously provided me, both scientifically and otherwise.

I would also like to acknowledge the financial support received from the reaserch grants „Dezvoltarea oportunităţilor oferite doctoranzilor pentru traiectorii flexibile în cariera de cercetare” POSDRU- 6/1.5/S/25, PNCDI II: “SIR – RESDEC Sistem de Întrebare-Răspuns în limbile Română şi Engleză cu Spaţii Deschise de Căutare”, PNCDI II: “eDTLR – Dicţionarul Tezaur al Limbii Române în format electronic” and PNCDI II: “INTELCHIM – Modelare şi

conducere automată utilizând instrumente ale inteligenţei artificiale pentru aplicaţii în chimie şi inginerie de proces”.

Abstract

One of the most relevant phenomena in natural language is that of variability, which means expressing the same thing using different surface representations. In order to address this issue, the notion of textual entailment was introduced (the notion of whether a text can be deduced from another). This thesis describes our contribution to the field of textual entailment.

Our approach is based on the notion of predicational semantics, as we believe that deep semantic understanding of natural language utterances can be more effectively deduced from the analysis of predicates and their arguments, and that deep semantic understanding is one of the best solutions for the problem of textual entailment. We have given a novel interpretation for the definition of textual entailment that builds upon our intuition. We have also described an algorithm for solving textual entailment on the basis of predicational semantics and argument structure unification. The approach described in this paper is both novel (there are, to our knowledge, no entailment systems based on predicational semantics) and effective, as our approach solves a large majority of entailment pairs (based on manual analysis of corpora) and the implementation based on our approach proved to have good results in evaluation campaigns.

The approach to solving textual entailment proposed in this thesis is language independent, given the apropriate lexical semantic resources available for that language. In order to extend existing resources, and to create new ones, we make use of our results with dictionary entry parsing, (mainly eDTLR), to propose ways of extending existing lexical semantic resources for Romanian, and for creating of new ones. The creation of these resources allow for the adaptation of our system to the Romanian language, without changing the TE algorithm we proposed.

The thesis is structured as follows: chapter 1 presents the general framework of textual entailment and describes the RTE challenges; chapter 2 gives the current state of the art for textual entailment; chapter 3 describes the theoretical basis of our approach; chapter 4 describes the implementation and results of our entailment system; chapter 5 discusses possible extensions to Romanian lexical-semantic resources based on eDTLR, together with the description of the parsing of the DLR, and chapter 6 discusses conclusions and future work, and provides an outline of the personal contributions of the thesis.

Table of Contents

1. Introduction

One of the most relevant phenomena in natural language is that of variability, which can be loosely defined as stating the same ideas in different ways. The issue of natural language variability needs to be addressed by various robust natural language processing applications, such as machine translation systems, summarization systems, question answering systems (QA) information retrievers (IR) or information extractors (IE), as they need to recognize the various forms in which information in the input and output texts is expressed. Most of the applications given above solve natural language variability, to a limited extent, by incorporating some form of shallow semantic parsing, but they do not employ a general framework for modelling variability in an independent manner (Dagan and Glickman, 2004).

The general solution for the issue of natural language variability is deep semantic understanding of natural language utterances, which would be used as an independent module that brings the input and output of the tools mentioned above to a common form (either as surface representations or as semantic descriptions). Obtaining deep semantic understanding of natural language utterances is one of the most difficult tasks in natural language processing (its difficulty has been compared to that of the Turing test (Bos and Markert, 2005)), and, as such, the common approach of extracting semantic representations for texts within each application is unfeasible. (Dagan and Glickman, 2004) describe a novel approach to removing the issue of natural language variability by defining the notion of textual entailment. According to (Dagan and Glickman, 2004), "textual entailment (entailment, in short) is defined as a relationship between a coherent text T and a language expression, which is considered as a hypothesis, H. We say that T entails H (H is a consequent of T), denoted by $T \Rightarrow H$, if the meaning of H, as interpreted in the context of T, can be inferred from the meaning of T." The entailment relation defined above is directional, since even if the meaning of one text may imply the meaning of another, the opposite does not always hold true.

The motivation of our thesis is to give a novel solution for the problem of textual entailment, based on the notion of predicational semantics and argument unification, which would lead to a deeper semantic understanding of natural language texts, and to provide a generalized, language independent framework for solving textual entailment, thus providing a robust entailment solver that can be attached to various natural language applications.

Since we consider that the one of the goals of textual entailment is to provide machine understanding for natural language at a level as close as possible to human understanding, we have also briefly examined (as an extension of the approach described in this thesis) a novel way of representing the semantic knowledge extracted from the text and the hypothesis, which would lead to deep semantic understanding of text, and is based on the idea of semantic triples represented in a graph format.

For a better understanding of the notion of textual entailment, we will refer to the Recognizing Textual Entailment Challenges (RTE), which will be described in detail in this section. Table 1 gives a sample of text – hypothesis pairs, taken from the Fifth Recognizing Textual Entailment (RTE-5) Challenge of 2009:

ID	Text	Hypothesis	Task	Solution
1	LOSAIL, Qatar (AFP) - Torrential rain caused the season-opening Qatar MotoGP to be cancelled on Sunday, leaving officials and teams in a frenzy before deciding to race on Monday instead at this floodlit desert venue. Monsoon-like conditions, accompanied by swirling winds, arrived just moments before Australia's Casey Stoner, on pole position, was due to lead defending world champion Valentino Rossi and the other riders away on the warm-up lap. "It's just unlucky with the weather," said Australian Ducati rider Stoner, the 2007 world champion, who was bidding for a third successive win here.	Valentino Rossi won the season-opening Qatar MotoGP.	QA	Contradiction
2	Euro MPs have voted overwhelmingly to cut the cost of texting and using the internet on mobiles abroad. The cap for a "roaming" text will fall to 11 euro cents (10p; 14 US cents), from about 29 cents on average today. The EU-wide caps, excluding VAT, will take effect in July. They cover text messages and data roaming services, such as checking e-mails while abroad. The current price cap of 46 euro cents per	A roaming text cost 46 euro cents.	QA	Contradiction

ID	Text	Hypothesis	Task	Solution
	minute for an outgoing voice call will also fall to 43 cents in July.			
82	Currently, there is no specific treatment available against dengue fever, which is the most widespread tropical disease after malaria. Sanofi Pasteur is collaborating with the Communicable Disease Center in Singapore and the Pasteur Institute in Vietnam to conduct these clinical studies in children and adults. "Controlling the mosquitoes that transmit dengue is necessary but not sufficient to fight against the disease. A safe and effective vaccine has been long awaited to prevent dengue epidemics," said Professor Leo Yee Sin, director of the Communicable Disease Center in Singapore. "Clinical studies in Singapore are critical steps to advance the development of a vaccine for the prevention of dengue in Asia. We are happy to contribute to scientific research that would benefit the entire region."	Malaria is the most widespread disease transmitted by mosquitoes.	QA	Unknown
207	The explosion happened at about 9:45pm local time, at the 'Rizk Plaza' shopping mall. It is understood from local media that a bomb was placed in the underground parking area below the complex, although there has been no official confirmation from security services. Sources on the ground say that the actual commercial center occupied the ground floor, with upper stories occupied by inhabited apartments. Located in the Metn range of mountains east of Beirut, Broummana is about half-hour's drive from the capital and is popular with tourists, particularly those from the Gulf states.	Beirut is near Broummana.	IE	Entailment
403	It is possible that it could help McCain earn the	Ronald Regan	IR	Unknown

ID	Text	Hypothesis	Task	Solution
	support of conservatives who have not always viewed him as aligning with the party on certain issues. At the same time, it could help to align McCain with former President Ronald Reagan, who attracted Republican and Democratic voters.	endorsed McCain.		

Table 1: Examples of text – hypothesis pairs from RTE-5

The table above shows text – hypothesis pairs taken from the test set of RTE-5 (http://www.nist.gov/tac/2009/RTE/). The first column gives the pair ID from the original corpus; the second and third columns are the pair itself (text and hypothesis), while the fifth column is the solution for the entailment problem of the given pair. Column four represents the type of natural language task from which the pairs were derived. There are three possible solutions for entailment pairs: ENTAILMENT, which means that the text entails the hypothesis, CONTRADICTION, denoting that the hypothesis contradicts some part of the text, and UNKNOWN, which means that no pertinent judgment can be drawn for the truth value of the hypothesis on the basis of the information in the text.

The following section gives a series of operational definitions for textual entailment, which are needed for formalizing the original definition given above to the point where it can be used in computational approaches, and section 1.2 gives a detailed description of the RTE challenges, which we consider relevant because of the significant impact they had on the textual entailment problem.

1.1. Operational Definitions for Textual Entailment

Even though the definition for textual entailment given above is complete and correct, it is too abstract to be directly used in practical applications of Natural Language Processing. Therefore, there have been a number of adaptations of the initial definition for textual entailment, in order to make it more practical for real life implementations, as can be seen in the list of definition variants given below:

- We say that a text T entails a hypothesis H if, typically, a human reading T would infer that H is most likely true. (Dagan et al., 2005). This version of the TE definition is aimed at human annotators, and is used for creating gold corpora for the task of textual entailment.

- T entails H if there exists a subgraph of XDG_T (the syntactic graph of T) that is in an isomorphism relation with XDG_H
- T entails H if there exists a sequence of transformations applied to T such that H can be obtained with an overall cost below a certain threshold, empirically estimated on the training data
- T entails H, if H does not introduce new information compared to T, or the combined model T+H is not more informative then T
- T entails H if all the atomic propositions in H are supported by at least one atomic proposition in T

Our intuition for solving textual entailment was also modelled within an operational definition. Given a text T and a hypothesis H, we say that T entails H, denoted by T→H, if and only if:

- Predicate matching: each of the predicates in H is entailed by at least one predicate in T. We say that a predicate $p \in T$ entails a predicate $q \in H$, denoted $p \rightarrow q$ if q is a consequence of p, or p and q are synonyms, or q is a subevent of p;
- Argument matching: given alignments in T for all of the predicates in H (we consider a predicate p in T aligned to a predicate q in H if there is an entailment relation between p and q), entailment holds if and only if the arguments of the aligned predicates are in an entailment relation. Two arguments are in an entailment reation if they both refer to similar entities (e.g., the heads of the arguments are synonyms, or the arguments are in a *part-of* relation, etc.), and if the unification of their feature sets is successful and is equal to the feature set of that argument in the text. Formally, if $p \in T$ and $q \in H$ and $p \rightarrow q$ and $\arg(p) = \langle a, b \rangle$, $\arg(q) = \langle a', b' \rangle$, then entailment holds if and only if $a \rightarrow a'$ and $b \rightarrow b'$. Argument entailment is defined as the result of the unification between the argument feature structures in H and T (as described in chapter 3 in greater detail).

The advantage of our operational definition of textual entailment is more semantically focused than previous versions, which leads to a deeper semantic understanding of the text and the hypothesis. Because of this we consider that our operational definition is closer to the original definition than many previous versions.

1.2. Recognising Textual Entailment Challenge (RTE)

The main proving ground for the various systems and methods of determining textual entailment is the *Recognising Textual Entailment Challenge* (RTE), which was initially put forth in 2005 by PASCAL (Pattern Analysis, Statistical Modelling and Computational Learning - http://www.pascal-network.org/) - the European Commission's IST-funded Network of Excellence for Multimodal Interfaces. Starting with 2008 (RTE-4), the RTE challenges were held within the Text Analysis Conference (TAC - http://www.nist.gov/tac/). TAC is a series of evaluation workshops which encourage research in Natural Language Processing by providing large test collections, common evaluation procedures and tools and opportunities for researchers to compare and share results.

The RTE Challenge has grown in time, increasing in complexity with each edition. Firstly, the size of the text in the entailment pairs grew steadily, from just one or two sentences in RTE-1, to an entire paragraph in RTE-4; RTE-5 went one step further, by using text which was not guaranteed to be grammatically correct.

With regards to the type of entailment pairs, RTE-1 had datasets created using such applications as Information Retrieval (IR), Comparable Documents (CD), Reading Comprehension (RC), Question Answering (QA), Information Extraction (IE), Machine Translation (MT), and Paraphrase Acquisition (PP). Starting with RTE-2, datasets consisted only of pairs extracted using Information Retrieval (IR), Information Extraction (IE), Question Answering (QA), and multi-document summarization (SUM), so as to create more realistic test data. By RTE-5, only Information Retrieval (IR), Information Extraction (IE) and Question Answering (QA) techniques were being used to create test pairs for the textual entailment engines.

Another innovation of the RTE Challenge was the introduction of pilot tasks, in RTE-3, RTE-5 and RTE-6. The RTE-3 pilot task consisted of having to classify the test pairs in three clusters: positive entailments, contradictions and unknown entailment. The main task of RTE-4 and RTE-5 became twofold: the initial 2-way task, first defined in RTE-1 and the new 3-way task, as described in the RTE-3 pilot task. For RTE-5, organizers created a new pilot task, focusing on extracting, from a collection of texts, those sentences that entail a given hypothesis. In RTE-6, the pilot task of the previous challenge became the new main task; in addition to this, a new pilot task was defined, the knowledge base population task.

1.2.1. First Recognising Textual Entailment Challenge[1]

According to (Dagan et al., 2005), "the RTE task is defined as recognizing, given two text fragments, whether the meaning of one text can be inferred from (entailed by) the other. This application independent task is suggested as capturing major inferences about the variability of semantic expression which are commonly needed across multiple applications." Based on the work of (Dagan and Glickman, 2004), the task of recognizing textual entailment is defined as deciding whether given two fragments of text, the meaning of one can be deduced from the meaning of the other.

In order to create a framework for solving textual entailment, the organizers created a data set consisting of text-hypothesis pairs using snippets of text extracted from news corpora. The entailment value of these pairs was then manually annotated, and then separated into training and test data sets. The goal of the participating entailment systems was to automatically decide whether entailment holds for the given pairs, and the results were then compared to the manually obtained gold standard. The data set was populated with text-hypothesis pairs corresponding to different textual processing applications, such as information retrieval (IR), comparable documents (CD – sentence alignment over similar documents), reading comprehension (RC), question answering (QA), information extraction (IE), machine translation (MT) and paraphrase acquisition (PP). The text-hypothesis pairs were selected in such a way as to highlight possible cases of success or failure for each of these applications, in order to better judge the effect an entailment system might have over their output. Also, the phenomena modelled by the entailment pairs range from simple lexical overlap to syntactic variability, logical deduction and world knowledge. The pairs have also been specifically chosen so that the

[1] http://pascallin.ecs.soton.ac.uk/Challenges/RTE/

distribution of the True and False examples is balanced (50%-50%), which does not apply to real world situations.

Since this was the first instance of the RTE challenge, the task definition, the evaluation methods and the corpus generation procedures were not yet mature, and much of the organizers' effort went into defining the methodologies that underlie the challenge. Because of this, this first challenge was aimed at a more exploratory setting, rather than a competitive one, and attempted to set a series of baselines, in terms of system performance and corpus generation. The table below gives the results of the top scoring systems in RTE-1 (based on the results reported in (Dagan, Glickman and Magnini, 2005)).

Submission	**Accuracy**	**Method**
(Pérez and Alfonseca, 2005)	0.7	Word overlap
(Delmonte et al., 2005)	0.606	WordNet, syntactic matching, logical inference
(Bayer et al., 2005)	0.586	Statistical lexical relations
(Glickman et al., 2005)	0.586	Statistical lexical relations

Table 2: Top scoring systems from RTE-1

The most basic method used for solving textual entailment in this setting was lexical overlap over words, lemmas, stems or parts of speech, using some sort of weighting (most commonly, inverse document frequency). More complex approaches used various relations between words, such as statistically derived relations or the WordNet taxonomy. Some systems used even more advanced methods, such as the degree of syntactic match, world knowledge or logical provers over semantically enriched representations. The decision mechanisms are also varied, and include probabilistic models, probabilistic machine translation models, supervised learning methods or specific scoring mechanisms, such as rules. Interestingly, system complexity does not fully correlate to performance, as some of the best results are obtained by rather naïve lexical-based systems. The RTE-1 challenge was quite successful, with 17 teams taking part.

1.2.2. The Second PASCAL Recognising Textual Entailment Challenge[2]

"The main goal of the second challenge was to support the continuation of research on textual entailment" (Bar-Haim et al., 2006). The general outline of the RTE-2 task was similar to that of RTE-1, in the sense that the purpose of the participating systems was to determine whether a hypothesis H is entailed by a text T. The data set was very different, as the organizers wanted to provide more realistic T-H pairs, based on outputs extracted from existing systems. The data collection and annotation process was also improved, as the guidelines for collection and annotation were improved (the data was cross-annotated by three different annotators).

The RTE-2 data set consists of 1600 entailment pairs, built using the basic setting of RTE-1. The data set was split into two 800 pair sets, for training and testing. In terms of application setting, the organizer only chose four of the original seven applications from RTE-1: IR, IE, QA and multi-document summarization (CD in RTE-1). Within each application setting, annotators selected both positive and negative examples, in equal number; the various application settings are equally represented in the corpus.

The main task of RTE-2 was classification of entailment pairs as either positive or negative cases; the evaluation measure used was accuracy (percentage of correctly classified pairs). A secondary task consisted of ranking pairs according to their entailment confidence; this means that the first ranked pair would be the one for which entailment is most certain, while the last pair is the one for which the "no entailment" judgement was most certain. The top scoring systems are given below (in the table below, and in subsequent tables, BK denotes Background Knowledge):

Submission	Accuracy	Method
(Hickl et al., 2006)	0.7538	Lexical relations, subsequence overlap, syntactic matching, SRL, web statistics, ML classification
(Tatu et al, 2006)	0.7375	Lexical relations, logical inference, BK
(Zanzotto et al., 2006)	0.6388	Lexical relations, syntactic matching, web statistics, ML classification

[2] http://pascallin.ecs.soton.ac.uk/Challenges/RTE2/

Submission	Accuracy	Method
(Adams, 2006)	0.586	Lexical relations, ML classification

Table 3: Top scoring systems from RTE-2

The results for RTE-2 are considerably higher than those for RTE-1, with top accuracies fro RTE-2 more than 10% above those for RTE-1. The important thing about the results of RTE-2 is that they correlate with the complexity of the method, which is to say that more complex systems perform better on this data set. However, simple lexical systems can still achieve an accuracy of over 60% (Bar-Haim et al., 2006). Also, the average system complexity grew, as did the number of participants, proving the appeal of the task.

1.2.3. The Third PASCAL Recognising Textual Entailment Challenge[3]

“RTE-3 followed the same basic structure of the previous campaigns, in order to facilitate the participation of newcomers and to allow "veterans" to assess the improvements of their systems in a comparable test exercise. Nevertheless, some innovations were introduced, on the one hand to make the challenge more stimulating and, on the other, to encourage collaboration between system developers” (Giampiccolo et al., 2007). Among the innovations are a limited number of entailment pairs for which the text was longer (up to a paragraph long), in order to make the scenario more similar to real life applications; it is worth noting, however, that the majority of the entailment pairs still had relatively short texts. Another innovation was the introduction of a resource pool, where participants could share the resources used. This was motivated by the conclusion drawn at the end of RTE-2, that modelling entailment requires large knowledge sources corresponding to different types of entailment reasoning. Also, entailment systems use various NLP tools, and thus the portal serves as a place for publicising and tracking resources, and for reporting on their usefulness.

As in the RTE-2 challenge, the RTE-3 corpus consisted of 1600 entailment pairs, equally divided into training and test sets. The pairs were drawn from the same application settings (IE, IR, QA, SUM) and the numbers of positive and negative examples were similar.

[3] http://pascallin.ecs.soton.ac.uk/Challenges/RTE3

RTE-3 also saw the introduction of a pilot task set up by the US National Institute of Standards and Technology (NIST)[4]. The purpose of the pilot task is to explore two notions for that are closely related to textual entailment: separating unknown cases from contradictions and providing justifications for system decisions. The first subtask requires that the entailment systems make more precise distinctions by making a three way decision between YES, NO and UNKNOWN, thus distinguishing between a hypothesis over which no significant judgement can be made and a hypothesis that explicitly contradicts the text. The second subtask was aimed at exploring ways in which users of entailment systems can receive justification for the entailment decision, as end users are unlikely to trust a result they cannot test. The pilot task used the existing infrastructure of the RTE-3 challenge, by expanding the data sets with annotation for the three-way task. The results for both the main and pilot tasks are given below (BK is the acronym for Background Knowledge).

Submission	Accuracy	Method
(Hickl and Bensley, 2007)	0.8	Lexical relations, subsequence overlap, logical inference, ML classification, anaphora resolution, BK
(Tatu and Moldovan, 2007)	0.7225	Lexical relations, logical inference, anaphora resolution, BK
(Iftene and Balahur-Dobrescu, 2007)	0.6913	Lexical relations, syntactic matching, BK
(Adams et al., 2007)	0.67	Lexical relations, subsequence overlap, web-based statistics, ML classification

Table 4: Top scoring systems from RTE-3 main task

Submission	Accuracy	Method
(Hickl and Bensley, 2007)	0.73	Lexical relations, subsequence overlap, logical inference, ML classification, anaphora resolution, BK
(Tatu and Moldovan,	0.71	Lexical relations, logical inference, anaphora resolution,

[4] http://nlp.stanford.edu/RTE3-pilot/

Submission	Accuracy	Method
2007)		BK
(Chambers et al. 2007)	0.59	Lexical relations, syntactic matching, logical inference, ML classification, anaphora resolution
(Iftene and Balahur-Dobrescu, 2007)	0.57	Lexical relations, syntactic matching, BK

Table 5: Top scoring systems from RTE-3 pilot task

1.2.4. TAC 2008 Recognizing Textual Entailment (RTE) Track[5]

On the basis of the positive feedback received from researchers interested in the RTE challenge, the organizers of the RTE challenges introduced a series of new elements, in order to keep the task affordable and stimulating (Giampiccolo et al. 2008).

The first major change in the RTE challenge was the decision to join efforts with the National Institute of Standards and Technology[6], which proposed the pilot task in RTE-3 and CELCT, which took part in all the previous RTE challenges; thus, the RTE challenge became a track of the Text Analysis Conference[7]. The other major innovation was the introduction of the RTE-3 pilot task as a main task for RTE-4.

For the RTE-4 challenge, participants were given 1000 text pairs, called Text and Hypothesis, for which they were required to determine whether T entailed H. Textual entailment is defined as a directional relation between two text fragments – T, the entailing text and H, the entailed text – so that a human being, with common understanding of language and common background knowledge, can infer that H is most likely true on the basis of the content of T (Giampiccolo et al. 2008). Unlike previous challenges, the entailment systems had to make a three way decision, by separating the cases for which there was no entailment by whether the truth of H was contradicted by T or it remained unknown on the basis of T. In short, the three-way task requires the system to decide whether:

---

[5] http://www.nist.gov/tac/2008/rte/

[6] http://www.nist.gov

[7] http://www.nist.gov/tac/

- T entailed H - in which case the pair was marked as ENTAILMENT
- T contradicted H - in which case the pair was marked as CONTRADICTION
- The truth of H could not be determined on the basis of T - in which case the pair was marked as UNKNOWN

The test set contained 1000 entailment pairs (300 each for IE and IR, as they were considered more difficult on the basis of previous evaluations, and 200 each for SUM and QA). 50% of the pairs were in an entailment relation, 35% were unknown and 15% were contradictions. On the basis of the three way challenge, results for the two-way challenge can also be extracted. The results for the RTE-4 challenge are given below.

Submission	**Accuracy for the 2-way task**	**Accuracy for the 3-way task**
(Bensley and Hickl, 2008)	0.746	-
(Iftene, 2008)	0.721	0.685
(Wang and Neumann, 2008)	0.706	0.614
(Li et al., 2008)	0.659	0.588

Table 6: Top scoring systems from RTE-4

Compared to the RTE-3 results, the top systems in RTE-4 had lower results. This may be due to the fact that the training and test sets were not developed concurrently.

1.2.5. PASCAL Recognizing Textual Entailment Challenge (RTE-5) at TAC 2009

The RTE-5[8] track at TAC 2009 (Bentivogli et al. 2010) continues the previous RTE Challenges that have aimed to focus research and evaluation on the underlying semantic inference task. The main structure of the RTE-5 challenge is mainly the same as that used in the RTE-4 campaign, but the data sets used had two significant changes: 1) the average length of the Texts was greater, and 2) texts came from a variety of sources and without any additional corrections or simplifications as compared to the source documents. These changed significantly increased the difficulty of the entailment task, but the reason behind the decision was that such texts are closer to real life applications of entailment.

[8] http://www.nist.gov/tac/2009/RTE/

The RTE-5 data set consists of 1200 pairs, of which half are used for training and the other half for testing. The pairs were drawn from the same application settings as those for previous challenges, with the exclusion of the summarization setting (entailment in the summarization setting was defined as a new pilot task). Also, the participants were required to submit ablation tests (runs of the entailment systems with components deliberately excluded) in order to judge the relevance of the knowledge sources employed by various systems.

The RTE-5 challenge also brought a new pilot task, defined by the organizers as solving textual entailment in a summarization setting and concerning the extraction of texts from a series of newspaper articles that yielded positive entailment for a given set of hypotheses. The difficulty of the task is twofold: first, the texts are not modified in any way as compared to the original source, so they may contain spelling errors, sentences with grammar errors, abbreviations and contractions, etc. The second problem is that there are a large numbers of candidate pairs, as for every one of the nine topics there are approximately ten hypotheses, and for every hypothesis in a topic the number of candidate pairs is equal to the number of sentences. This leads to a very large search space, and the problem to reduce it becomes very important.

Submission	Accuracy for the 2-way task	Accuracy for the 3-way task
(Iftene and Moruz, 2010)	**0.6833**	**0.735**
(Wang et al, 2010)	0.6367	0.685
(Ferrandez et al., 2010)	0.6	-
(Malakasiotis, 2010)	0.575	-
(Mehdad et al, 2010b)	-	0.6617

Table 7: Results for the RTE-5 main task

Submission	Precision	Recall	F-measure
(Mirkin et al, 2010, RTE)	0.4098	0.5138	0.4559
(MacKinlay and Baldwin, 2010)	0.4294	0.38	0.4032
(Mehdad et al, 2010a)	0.2254	0.6475	0.3344

(Iftene and Moruz, 2010)	**0.5112**	**0.2288**	**0.3161**

Table 8: Results for the RTE-5 pilot task

Even though the data sets for the RTE-5 challenge are significantly more complex than those of the previous challenges, the average score of the system has improved, and the top scores are similar, thus proving that entailment systems had reached maturity. Also, the introduction of the new pilot task offered a new perspective on the idea of textual entailment in a real life application, and highlighted the difficulties that arise when adapting systems built for entailment in an artificial setting.

1.2.6. PASCAL Recognizing Textual Entailment Challenge (RTE-6) at TAC 2010

The feedback provided through the RTE challenges by the NLP community has given the opportunity for the application of RTE systems to various settings and has pointed the RTE community towards more realistic scenarios. In particular, the RTE-5 Pilot Search Task represented a step forward, as for the first time textual entailment recognition was performed on a real text corpus. Furthermore, it was set up in the Summarization setting, attempting to analyze the potential impact of textual entailment on a real NLP application.

According to the RTE-6 guidelines[9] the goals of the challenge are:

- to advance the state of the art in RTE, by proposing a data set which reflects the natural distribution of entailment in a corpus and presents all the problems that can arise while detecting textual entailment in a natural setting, such as the interpretation of sentences in their discourse context;
- to further explore the contribution that RTE engines can make to Summarization applications. In a general summarization setting, correctly extracting all the sentences entailing a given candidate statement for the summary (similar to Hypotheses in RTE) corresponds to identifying all its mentions in the text, which is useful to assess the importance of that candidate statement for the summary and, at the same time, to detect those sentences which contain redundant information and should probably not be included in the summary. Furthermore, if

[9] http://www.nist.gov/tac/2010/RTE/

automatic summarization is performed in the Update scenario (where systems are required to write a short summary of a set of newswire articles, under the assumption that the user has already read a given set of earlier articles) it is important to distinguish between novel and non-novel information. In such a setting, RTE engines which are able to detect the novelty of H's can help Summarization systems filter out non-novel sentences from their summaries.

In the new setting of RTE-6, the challenge of identifying textual entailment changes in the following way: given a corpus, a hypothesis H and a set of candidate entailing sentences retrieved from the corpus using Lucene, the systems need to identify all the sentences among the candidates that entail the hypothesis. This setting for the entailment problem is very different from the one originally proposed in the previous RET challenges; it is concerned with entailment within a corpus, with both the text and the hypothesis needing to be interpreted in the larger setting of the available knowledge, and it reflects the natural distribution of entailment (which is to say there are a lot more negative examples than positive ones).

In the RTE-6 main task scenario (Bentivogli et al., 2011), the test set is defined as a set of topics, each containing document clusters. For each of the topics, a set of up to 30 standalone sentences is extracted, and those sentences will serve as hypotheses for the topic. For each H in the topic a Lucene query is run (all the words in the hypothesis connected by the OR logical operator), and the top 100 ranked sentences are selected as candidates (it has been empirically determined that 80% of all the entailing sentences in the corpus are found in this manner). Based on the Main Task, a subtask is also defined. It is focused on Novelty Detection, which means that RTE systems are required to judge whether the information contained in each H is novel with respect to (i.e., not entailed by) the information contained in the corpus. If entailing sentences are found for a given H, it means that the content of the H is not new; in contrast, if no entailing sentences are detected, it means that information contained in the H is novel. Although the Novelty Detection Task has the same structure as the Main Task, it is separated out as a subtask to allow participants to optimize their RTE engines differently (i.e., for novelty detection).

Submission	**Precision**	**Recall**	**F-measure**
(Jia et al, 2011)	0.6857	0.3693	0.4801

Submission	Precision	Recall	F-measure
(Majumdar and Bhattacharyya, 2011)	0.5343	0.4286	0.4756
(Tateishi and Ishikawa, 2011 IKOMA)	0.3971	0.5143	0.4481
(Kouylekov et al., 2011)	0.4346	0.4603	0.4471

Table 9: Results for the RTE-6 main task

Submission	Precision	Recall	F-measure
(Jia et al, 2011)	0.7239	0.97	0.8291
(Tateishi and Ishikawa, 2011)	0.7944	0.85	0.8213
(Pakray et al, 2011)	0.8058	0.83	0.8177
(Volokh et al., 2011)	0.735	0.86	0.7926
(Iftene and Moruz, 2011)	**0.7328**	**0.85**	**0.787**

Table 10: Results for the RTE-6 novelty detection subtask

1.3. Conclusions

In this section we have provided an overview of the RTE challenges, starting with the first, introduced in 2005, up to the latest, in 2010. Through the evolution of the data sets, the tasks and the results from year to year, the evolution of textual entailment systems can be followed. Because of this, and because the RTE challenge provides a unified testing ground for textual entailment engines, its importance cannot be understated.

As can be seen from the large number of participants (the number of competitors has grown steadily until RTE-5), the interest of the NLP community for this problem is significant, which leads us to believe that improved textual entailment engines are still needed.

The rest of the thesis is structured as follows: chapter 2 describes the state of the art in textual entailment, chapter 3 describes the method proposed by us for solving textual entailment, chapter 4 gives evaluation results for the system we have built for solving textual entailment, along with a brief description, chapter 5 discusses the adaptation of the entailment system we have built to the Romanian language, and conclusions are given in chapter 6.

2. State of the Art in Textual Entailment

The very first approaches to solving textual entailment were based on word overlap (Herrera, 2005), statistical lexical relations (Bayer et al., 2005), WordNet similarities (Herrera, 2005), syntactic matching (Delmonte et al., 2005), world knowledge (Bayer et al., 2005), logical inference (Akhmatova, 2005), inversion transduction grammars (Wu, 2005), (Jijkoun and de Rijke, 2005) or edit distance between parsing trees (Kouylekov and Magnini, 2005). Eventually, systems evolved to using a series of new directions, which were starting to turn away from the initial shallow approaches, with systems making use of semantic role labelling (Hickl et al., 2006), Machine Learning classification (Inkpen et al., 2006 and Kozareva, 2006), using of background knowledge (Tatu et al., 2006), acquisition of entailment corpora (Hickl et al., 2006), and rule based entailment (Iftene, 2009).

2.1. Lexical Representation

One of the first successful systems for recognizing textual entailment was an application of the BLEU algorithm, described in (Pérez and Alfonseca, 2005). The BLEU (BiLingual Evaluation Understudy) was first described in (Papineni et al., 2001), and was initially used for ranking machine translation systems. It provides a score on the validity of a given translation candidate (usually obtained automatically, by means of machine translation) on the basis of its similarity to a manually obtained translation of the same source text. In order to cope with variability, the candidate is compared to a set of manually obtained translations, and the top score is chosen; BLEU boosts the similarity score for a given candidate in the case of matching n-grams. It is worth noting that after computing the score of the co-occurring n-grams, high scores of very short candidates are penalized on the basis of their length, as it is considered that a translation variant should not be significantly different in length than the text. This drawback, coupled with the fact that the BLEU algorithm is inherently symmetrical, make the adaptation for the task of recognizing textual entailment not trivial.

The application of the BLEU algorithm for textual entailment consists of comparing the entailing text (T) and the hypothesis (H); according to the score provided by BLEU, the entailment is then judged to be true or false. According to (Pérez and Alfonseca, 2005), using T as the reference and H as the candidate is best, as T is usually longer and therefore contains more information, which helps the BLEU processing, as the quality of the references is very important. This adaptation, which essentially consists of only comparing part of T against H, circumvents both problems outlined above with the usage of the BLEU algorithm.

The very good result obtained by the system (it was ranked first in RTE-1, with an accuracy of over 70%) proves the potential of the approach. However, this method is not without drawbacks: it does not make any use of syntactic information, and in the case of hypotheses that have many words in common with the texts, the answer will always be positive entailment. Another significant drawback is that this approach does not work well on long texts (such as those used in the RTE-5 challenge), as it becomes very difficult to extract significant references that are both relevant and valid for determining entailment (not extracted from different sentences for example).

With the growing complexity of the task of recognizing textual entailment, lexical based approaches have grown significantly more complex, by making use of wider and wider lexical-semantic resources. A brief overview of some of the most significant approaches in recent years is given below.

BLUE-Lite (Clark and Harrison, 2011) is a derivative of the RTE5 system BLUE (Clark and Harrison, 2010), and is characterized by the following features: **1.** The first step taken to compare T and H is to parse them with SAPIR, a broad coverage chart parser (Harrison and Maxwell, 1986), and to transform both T and H into bags of words which also include multiwords present in WordNet; all the functional words are ignored, as well as some light semantic verbs, given by the authors. **2.** This approach could thus be described as "knowledge-based lexical entailment" that makes use of a series of relations from WordNet, such as the WN morphosemantic database, that relates nouns to verbs (`build#v1 -agent→ maker#n1`), the "pertains-to" relation (`rapidly#r1 ←pertains-to→ quick#a1`), the "similar-to" relation for adjectives (`nice#a1 ←similar-to→ pleasant#s2`), the "part-of" relation, and the "substance-of" relation. The approach also uses the DIRT paraphrase database. **3.** In order to account for coreference and discourse context, the system attempts to match those words in H that cannot be matched to words in T to words in the previous sentence. **4.** Some topics were intrinsically harder to find entailments for than others. To account for this, the system gradually reduces the restrictions on the matching of the words in H, by allowing for at most 2 words to be matched with sentences preceding T. The most important thing to note is that, unlike BLUE, BLUE-Lite does not use of any structural (parse-based) information in entailment decisions, but still obtains reasonably good results (in relative terms), with an f-score of more than 44% on the RTE-6 test set[10].

(Majumdar and Bhattacharyya, 2011) describes a lexical based system for solving entailment that was used for the RTE-6 competition, which is similar to the one proposed by (Clark and Harrison 2011), yielding good results (47%). The system draws on lexical knowledge from WordNet (for measuring the length of the path between words) and VerbOcean (for determining entailment between verbs), and uses the Stanford named entity recognition system, enhanced with acronym detection. The system is built to take into account various types of coreference, but the reported tests do not include the coreference recognition module.

(Perini, 2011) is a lexical based entailment system used in RTE-6, and it is an evolution of the system described in (Perini, 2010) that uses directional text relatedness conditions for detecting textual entailment between sentences. The system is centred on computing the text relatedness score between T and H, with respect to both T and H. In general, the directional relation of entailment can be described as $sim(T,H)_H > sim(T,H)_T$ (the similarity between T and H with regards to H is greater than the similarity between T and H with regards to T), if T entails H, where `sim` is the measure of similarity between H and T. The approach is based on the method proposed by (Tătar et al., 2009), which describes a directional approach to solving textual entailment, based on the notion of lexical-semantic similarity.The system described in (Perini, 2011) uses WordNet to compute similarity between words, by assigning scores on the basis of the relation detected using WN (if two words are in the same synset, the score is 0.7). This approach yielded middle range results in RTE-5, and comparatively high results in RTE-6 (40% f-measure).

(Tateishi and Ishikawa, 2011) proposes a new method for identifying textual entailment by making use of local-novelty detection. The method is based on determining whether a hypothesis H is local-novel. 'H being local-novel' can be described as the fact that the information described by H first appeared in T_H , and T_H denotes the text (T) that entails H.

[10] It is important to note that results in different RTE competitions cannot be compared due to differences in datasets.

Based on this description, it follows that H is local-novel, the first possible entailing text has to be T_H, and therefore any T that appeared before it has to be non-entailing. The authors define six rules for determining the local-novel feature; on the basis of this feature, they provide a threshold of similarity, which describes the minimum required similarity of T and H in order to obtain entailment.

(Blake et al., 2011) explores the impact that semantics alone has on the accurate detection of entailment, in order to better judge the impact of this type of component on an entailment system. The authors considered various lexical-semantic knowledge bases and resources, and eventually used WordNet and YAGO. The reason for using YAGO, developed as part of the YAGO-NAGA system, is that it seems to be more comprehensive in terms of names than WordNet. YAGO is a bootstrapped ontology, which uses information given in WordNet to infer further information from the web. The disadvantage of this approach as compared to manually constructed resources such as WordNet is the fact that automatic bootstrapping can lead to errors. Names of persons and organizations, locations and numbers are compared on the basis of the results obtained by the lexical knowledge bases, and a similarity score is computed. The rest of the words (those not recognized as NEs or co-reference, for example) are matched lexically, on the basis of the lemma form, or semantically, by means of synonymy in WordNet.

(Pakray et al. 2010) describes a system based on the composition of six lexical matching methods: WordNet based unigram match, bigram match, longest common sub-sequence, skip-gram (any combination of *n* words in the order in which they appear in the text, allowing for arbitrary gaps), stemming and named entity matching. The authors then trained a number of classifiers on the features described above, which they then used as a voting system for determining entailment. Since the classifiers had to be in complete agreement in order for the system to make a positive decision, the best results are obtained when only two classification methods are used.

(Breck, 2010) proposes a novel approach to detecting entailment, on the basis of the observation that if T entails H, it follows that all of the information in H must be contained within T. Therefore, the author proposes a system that can determine whether all the units of information in H can be found in T; if there exists at least one unit of information in H that cannot be found in T, the result for that entailment pair is Unknown. If all the units of information in H can be found in T, it is also possible that the result can be Contradiction, and so a second, simpler, system is used for determining contradictions. The units of information taken into account are words, but the author only takes into account open-class words such as common nouns, proper nouns, verbs, adjectives, adverbs, and numbers. Based on experiments carried out by the author, it has been proven that the system gains most from using common nouns, proper nouns, and numbers. The matching methods employed are exact string match, edit distance, acronyms, and lexicon based matching (WordNet and Dekang Lin's automatically derived thesaurus, which experiments showed did not have high enough accuracy[11]).

(MacKinlay and Baldwin, 2010) proposes a system specifically designed for solving the task of textual entailment in a summarization setting, on the basis of variants of standard document comparison techniques. The system computes the lexical overlap of candidate texts for a given hypothesis by means of cosine similarity on bag of words features, weighed by inverse document frequency. The system also uses WordNet for determining synonymy and derivationally related forms, and the results obtained are above the median for the RTE-5 pilot task.

This subsection described the most successful approaches to solving textual entailment by using lexical and lexical-semantic features. Even though this type of approach is hindered by the lack of more complex analyses, such as syntactic parsing for example, the results described are quite good, especially for the summarization setting of textual entailment. This proves that, in a significant number of entailment cases, the measure of lexical similarity is still significant.

2.2. Syntactic Graph Distance

Graph distance is a method that is widely recognized as a powerful tool for solving computer vision problems and for performing pattern matching (Bunke and Shearer, 1998), and was the method first used by (Pazienza et al., 2005) for creating a textual entailment system. Syntactic graph distance based approaches work by transforming the objects that need comparing (in the case of textual entailment, the text and the hypothesis) into graphs, thus transforming the entailment problem into a graph matching problem.

---

[11] http://www.cs.ualberta.ca/~lindek/downloads.htm

Since any natural language sentence can be represented by a syntactic graph, textual entailment can be reduced to a graph similarity problem, albeit with some particular properties (Pazienza et al, 2005): this problem of textual entailment is not symmetric, node similarity cannot be reduced to similarity between node labels (we cannot compare pre-terminal nodes in the syntactic trees), and linguistic transformations need to be taken into account. Therefore, on the basis of graph similarity, entailment can be defined as a non-symmetric transitive relation that can be defined as follows (Pazienza et al., 2005):

- T is syntactically subsumed by H (e.g., in H: [John is walking.] and T: [John is walking fast.], T contains an adverb, making the text more specific than the hypothesis).
- T is semantically subsumed by H (e.g., in H: [John runs home.] and T: [John sprints home], where the verb "run" is more general than the verb "sprint").
- The information in T is directly implied by the information in H (e.g., T: [John is snoring] and H: [John is sleeping.], the fact that John is snoring cannot take place unless John is sleeping).

The system described in (Pazienza et al., 2005) attempts to find the maximal subgraph within the syntactic representation of T that is isomorphic with the syntactic representation of H. This method is more advanced than the lexical similarity based approaches, and has the advantage of taking into account both the syntactic and the semantic levels. Graph similarity has been used by many systems, including those of (Katrenko and Adriaans, 2006), (Zanzotto et al. 2006), (Burchardt et al., 2007), (Ferrés and Rodríguez, 2007) and (Padó et al., 2008). More recent graph distance based approaches are given below.

(Mehdad et al., 2010b) proposes a textual entailment engine based on the notion of syntactic/semantic kernels. Syntactic kernels were first explored by the authors in (Zanzotto and Moschitti, 2006), where the basis of the approach to solving entailment was the alignment of syntactic subtrees from T and H by means of cross-pair syntactic kernels, which were then used to train an SVM classifier. Standard tree kernel functions, measure the similarity of two trees in terms of the number of tree fragments (subtrees or substructures) that they have in common, with the addition of so called "placeholders", which are the means used to track the movement of aligned words in the text and hypothesis. However, this first approach, even though it performed quite well on the RTE-1 dataset, is limited by the fact that two syntactically identical subtrees which have different leaves do not match even if the leaves, i.e. the words, are synonyms.

(Mehdad et al. 2010b) proposes the augmentation of the previous syntactic kernels with the use of semantic knowledge based on Wikipedia, thus creating an entailment system based on syntactic/semantic kernels. More specifically, Wikipedia was used to enrich the similarity measure between pairs of text and hypothesis (i.e. the tree kernel for text and hypothesis pairs), with a lexical similarity (i.e. the similarity between the leaves of the trees). The lexical similarity is defined as a proximity matrix, which is filled in by considering that words are semantically related if they co-occur frequently in Wikipedia articles. The Syntactic Semantic Tree Kernel (SSTK) is the augmentation of the Syntactic Tree Kernel defined in (Zanzotto and Moschitti 2006) with the similarity measure derived from Wikipedia. In order to reduce the search space, only the first 200,000 most visited articles in Wikipedia were used for extracting similarity between terms, and similarity was computed only for those words that actually appeared in the text-hypothesis pairs. The kernel used for the system is Max Similarity Kernel, though the authors also propose the Placeholder Kernel as an alternative. The MSK-SSTK combination was used in the system for RTE5, which achieved 66% accuracy in the 2 way recognition task, well above the median and 6% below the top scoring system in the competition. A more detailed description of the tree kernels employed is given in (Mehdad et al. 2010c). The coverage and usefulness of WordNet and of the British National Corpus (BNC is a balanced synchronic text corpus containing 100 million words with morpho-syntactic annotation) are also explored, and proven to be much less than those of Wikipedia, by quite a large margin (Wikipedia coverage is almost double than that of both WordNet and British National Corpus).

2.3. Tree Edit Distance Algorithms

One of the first uses of tree edit distance algorithms for textual entailment is the one proposed by (Kouylekov and Magnini, 2005), which describes a system based on a tree edit algorithm applied on the dependency trees of both the text and the hypothesis. According to (Kouylekov and Magnini, 2005), “T entails H if there exists a sequence of transformations applied to T such that we can obtain H with an overall cost below a certain threshold”. This is based on the intuition that pairs where the entailment relation holds have low transformation costs, in a similar fashion to the approach proposed in subsection 2.2. The difference in this type of approach is that instead of matching subtrees from the hypothesis to ones in the text, this mapping is a sequence of transformations applied to a syntactic tree, where each operation has an associated cost; as such, an entailment relation is confirmed if the overall cost of the transformation is below a certain corpus dependant threshold.

(Kouylekov and Magnini, 2005) proposes the following operations for transforming syntactic trees:

- **Insertion**: insert a node from the dependency tree of H into the dependency tree of T. When a node is inserted, it is attached to the dependency relation of the source label.
- **Deletion**: delete a node N from the dependency tree of T. When N is deleted, all its children are attached to the parent of N. It is not required to explicitly delete the children of N as they are going to be either deleted or substituted on a following step.
- **Substitution**: change the label of a node N1 in the source tree into a label of a node N2 of the target tree. Substitution is allowed only if the two nodes share the same part-of-speech. In case of substitution, the relation attached to the substituted node is changed with the relation of the new node.

Other approaches that used tree edit distance algorithms are those of (Katrenko and Adriaans, 2006), (Kozareva and Montoyo, 2006), (Harmeling, 2007), (Iftene and Balahur-Dobrescu, 2007), (Bar-Haim et al., 2008). More recent approaches that employ tree transformations are given below.

A novel approach to textual entailment is given in (Cabrio and Magnini 2010) and is based on the notion of specialized entailment engines that rely on monothematic entailment pairs. A monothematic pair is defined as a text-hypothesis pair in which a certain phenomenon relevant to entailment is highlighted and isolated. Monothematic pairs can be created on the basis of phenomena that are present in available entailment pairs, in order to identify their distribution within the entailment relation. The intuition of the authors is that if a system can correctly solve the underlying phenomena of entailment, it can correctly solve entailment in complex cases, and, therefore, a high accuracy on the monothematic pairs should imply a high accuracy in the general case. In order to test this, the authors used 60 entailment pairs from the RTE-5 test set examples) and a dataset composed of all the monothematic pairs derived by the first one. Two systems used in the RTE-5 competition were run on the separate datasets, in order to test the correlation between the proposed method over singular linguistic phenomena and the general case. In the case of EDITS (Mehdad et al., 2010a), it was found that the system does better on the monothematic corpus, while in the case of VENSES (Delmonte et al., 2009) it performed worse. The relevance of this approach lies in the fact that it is able to identify the strengths and weaknesses of different entailment systems by means of the correlation measure between results obtained on a standard corpus and those obtained on a monothematic corpus.

An extension to the classical tree edit distance approach is given in (Heilman and Smith 2010). This extension is motivated by the fact that, in some instances, the same information can be represented in significantly different ways, and the operations of insertion, deletion and substitution yield a very high cost. Standard tree edit distance is thus extended with the addition of some new operations and the modification of some existing ones. The editing operations used are `insert-child` (as the last child of the target node), `insert-parent` (add a node between the target node and its parent), `delete-leaf` (deletes a leaf node), `delete-&-merge` (deletes an internal node in the tree and adds all of its children to its parent), `relabel-node` (changes the lemma and part of speech of a node), `relabel-edge` (changes the edge label), `move-subtree` (removes a given subtree from its former position and adds it as the last child of a node), `new-root` (changes the root of a tree to a given node, and adds the old root as the last child of the new root) and `move-sibling` (moves a node to be either the first or last child of its parent). In order to reduce computational costs, the search space is reduced by means of restrictions to the application of the rules given above, and the search heuristic is based on tree kernels. To prove the viability of the approach, it has been tested on three tasks: recognizing textual entailment, paraphrase identification and answer selection. The training for the recognizing textual entailment setting was carried out over the RTE-1, RTE-2 and RTE-3 datasets, and testing over the RTE-3 dataset. The results obtained were competitive, and the method proposed shows how a tree edit based system can be improved by means of adding more types of editing operations.

The most successful application of tree edit distance to the task of recognizing textual entailment is that described by (Mehdad et al. 2010a), which is a development of the approach proposed in (Kouylekov and Magnini, 2006). The system which took part in the RTE Challenge is the EDITS package (Edit Distance Textual Entailment Suite), developed by the HLT group at

FBK. EDITS implements a distance-based approach for recognizing textual entailment, according to the assumption that the distance between T and H is a characteristic that separates positive entailment pairs from negative ones (EDITS is built for the 2 way entailment task). The distance is computed as the overall cost edit operations (i.e. insertion, deletion and substitution) needed to transform T into H. The edit distance approach used in EDITS has three main components: an edit distance algorithm (either string edit distance, token edit distance or tree edit distance), a cost scheme for the operations and an optional set of rules, which can be either manually added or extracted from external resources. The main sources of knowledge used for the RTE5 submissions are: i) a list of stopwords (the 572 most common English words), ii) WordNet (hyponymy and synonymy relations), iii) VerbOcean (”stronger-than” relation), and iv) Wikipedia (Latent Semantic Analysis (LSA) over Wikipedia, as a huge knowledge resource, between all possible node pairs (terms or lemmas) that appear in the dataset.).

For the RTE-6 challenge, where the task was significantly different from previous ones, the EDITS system was again used with good results, described in (Kouylekov et al. 2011). In addition to previous years, the authors experimented with: i) the use of lexical knowledge (in the form of entailment rules mined from different resources including Wikipedia, VerbOcean, WordNet, Lin’s Proximity and Dependency thesauri, ii) different performance optimization criteria, iii) the use of algorithms working at the level of syntactic representations of T and H, and iv) the combination of different algorithms.

2.4. Logical Inference

The use of logical inferences has also been a successful in terms of solving textual entailment. One of the first such approaches is that proposed by (Bos and Markert, 2005), which employs two methods. The first uses “several shallow surface features to model the text, hypothesis and their relation to each other”, as the authors expect “some dependency between surface string similarity of text and hypothesis and the existence of entailment” Similarity is computed by word overlap between the text and hypothesis, taking into account synonyms and derivations extracted from WordNet (Fellbaum, 1998).

The second approach uses deep semantic analysis, using state-of-the-art off-the-shelf inference tools, namely a theorem prover and a model builder. The system creates a fine-grained semantic representation for each text-hypothesis pair, and entailment is checked using two kinds of automated reasoning tools: a theorem prover and a model builder (both these tools work on inference problems stated in first order logic form). The entailment problem is solved by means of the theorem prover by attempting to find answers to tow conjectures: T implies H and T+H are inconsistent. Robustness comes from the idea that If H is entailed by T, the model for T+H is not informative compared to the one for T, and hence does not introduce new entities. Also, differences in domain sizes are relevant in determining entailment, as small differences denote positive entailment, while large differences denote negative entailment (the domain of T is defined as the trasitie and reflexive closure of the model of T).

In the course of the RTE challenges, there were a number of approaches that were either based on logical inference or that included some form of logical inference, such as those of (Raina et al., 2005), (Tatu et al., 2006), (Hickl and Bensley, 2007), (Tatu and Moldovan, 2007), (Clark and Harrison, 2008) and (Bergmair, 2008).

(Clark and Harrison, 2010) describes one of the most successful textual entailment engines that is based on logical inference. The idea behind BLUE (Boeing's Language Understanding Engine) is to convert the sentences from the text and the hypothesis into a logic-based representation, and then search to see if T implies (or contradicts) H. Compared to the previous versions of entailment systems proposed by the authors, the current system consists of two pipelined modules: the logic representation module and a new bag of words representation module. The basic components of the logic representation module of BLUE are a parser, a logical form (LF) generator, and final logic generator. Parsing is performed using SAPIR, a bottom-up, broad coverage chart parser. During parsing, the system also generates a logical form (LF), a semi-formal structure between a parse and full logic. The LF is a simplified and normalized tree structure with logic-type elements, generated by rules parallel to the grammar rules, that contains variables for noun phrases and additional expressions for other sentence constituents. The LF is then used to generate ground logical assertions of the form r(x,y), by applying a set of syntactic rewrite rules recursively to it. Entailment is recognized by either checking for subsumption between (logical) predicates and their arguments, or by using inference rules deduced from WordNet (on the basis of synonymy, hypernymy, and the similar, pertains and derivational links) and the DIRT database. The second module, the bag of word module, is only used in those cases where the logic representation module is unable to give a definite answer for a given entailment pair. This module searches for words in the text so that each word in the hypothesis subsumes a text word. Subsumption is computed on the basis of WordNet and DIRT, in a similar manner to that used in the logical representation module.

The system scored above the median in the RTE-5 main task, but it was still below the top system by more than 10%; this shows that this type of approach has significant drawbacks, a fact noted by the authors who decided to switch to a more robust approach, based on lexical semantics, for the RTE-6 competition.

2.5. Atomic Propositions

The atomic proposition based approach works on the idea of decomposing the text and the hypothesis into atoms and attempting to support atoms in the hypothesis with atoms in the text. The atoms in this approach are atomic proposition, which are minimal declarative statement that is either true or false, and whose truth value does not depend on any other atomic proposition.

The first application of the idea of atomic propositions is that of (Akhmatova, 2005), which uses an automated deduction system to compare atomic propositions. The proposed system is capable of recognizing basic entailments using syntactic and semantic information, and shows great promise given the addition of more knowledge, which would allow it to perform better on complex entailment pairs. (Akhmatova, 2005) provides the following example: the sentence "*Coffee boosts energy and provides health benefits*." can be split into two atomic propositions, namely "*Coffee boosts energy*." and "*Coffee provides health benefits*." In order to extract atomic propositions, the author uses an algorithm described in (Jurafsky and Martin, 2000); also, the author believes that deep syntactic and semantic analyses are vital to solving the problem.

(Hickl and Bensley, 2007), on the other hand, proposes an approach based on extracting discourse commitments from the text and the hypothesis. The methods used for extracting discourse commitments include sentence segmentation, syntactic decomposition, supplemental expressions, relation extraction, and coreference resolution. Once the sets of discourse commitments is identified, entailment can be deduced if each commitment extracted from the hypothesis is supported by at least one commitment in the text.

A more recent approach based on the notion of atomic propositions is the system proposed by (Ofoghi and Yearwood, 2010), *UB.dmirg*. To identify entailment relationships between texts and hypotheses, the system employs a term-based approach that analyzes the text and hypothesis at the lexical level and then produces relations at the text level (atomic propositions). Each sentence in the hypothesis and the text is parsed by means of a Link Grammar parser, and on the basis of the links extracted and a series of syntactic rules, propositions are extracted. Given atomic propositions, the system reports entailment when every proposition in the hypothesis is supported by at least one proposition in the text. Propositions are compared utilizing WordNet and FrameNet for estimating lexical coverage of the text on the hypothesis, and in the case of non grammatical sentences (where the Link Grammar parser does not return any parses), Levenstein distance is used as a fallback.

For the 2010 version of the *UB.dmirg* system, described in (Ofoghi and Yearwood, 2011), the authors based their system on the idea of machine learning over lexical-semantic features, a completely different approach from that of (Ofoghi and Yearwood, 2010). For this, it employs FrameNet (making use of a number of inter-frame relationships) and WordNet (using synonymy, antonymy and hypernymy) resources to extract event-based and semantic features from both hypothesis and text. The system also uses the longest common substring of lemmas when learning the entailment relationships. The features were then used to train three classifiers implemented in the WEKA library: i) K-Nearest Neighbour classifier, ii) Random Forest classifier, and iii) Bayesian Network classifier. The results obtained were not promising, as they fell well below the median.

Currently, the one of the most successful approach to solving textual entailment that uses atomic propositions (or similar structures) remains that of (Bensley and Hickl, 2008), further described in (Hickl, 2008) which is by far the best performing in the category and one of the most successful in general, with the top score in the 2008 RTE challenge.

2.6. Machine learning

Machine learning techniques were used to solve textual entailment since the introduction of the challenge, in RTE-1. The range of features used is wide, and includes lexical match, lexical-semantic match, semantic matching, syntactic matching, etc., and use various resources and theories such as the WordNet taxonomy (Miller, 1995), the VerbOcean semantic network

(Chklovsky and Pantel, 2004), the Discourse Representation Theory (Kamp and Reyle, 1993) for determining negation, etc. The learning algorithms used by the various systems were also numerous, including SVM (Joachims, 2002), C5.0 (Quinlan, 2000), binary classifiers like Bayesian Logistic Regression (BBR), etc. More recent approaches use machine learning algorithm repositories, such as Weka[12], in order to better judge the effect of the training method on the system performance. Machine learning techniques were also employed for automatically extracting entailment rules from existing knowledge bases, such as Wikipedia, or for extending existing rule repositories (Szpektor and Dagan, 2008).

The number of systems using machine learning techniques grew steadily, as the amount of available training data increased (most of the machine learning based systems use all of the available data from the RTE challenges). The approaches and results vary, depending on the manner in which authors combine and use attributes; also, since the data sets vary from challenge to challenge, the gain from using more than on at a time is not significant. Major machine learning based systems include those of (Inkpen et al., 2006), (Li et al., 2007), (Kozareva and Montoyo, 2006) and (Montejo-Ráez et al., 2007).

Some of the more significant approaches to solving textual entailment by means of machine learning are given below.

(Castillo, 2010a) describes a system that uses machine learning algorithms and a combination of datasets for the task of recognizing textual entailment and that was initially used for the RTE-4 evaluation campaign. The author attempts to solve entailment in two ways: by directly using a series of lexical-semantic features (the Levenshtein distance between each pair, a semantic distance based on WordNet and the Longest Common Substring) or by using the NER-pre-processing module to determine whether non-entailment is found between text and the hypothesis, on the basis of absence of Named Entities. The system was trained using the RTE-1, RTE-2, RTE-3 and RTE-5 datasets, taken separately and in various combinations. The testing was carried out over the RTE-4 test set, and the best results were obtained with the combination of the RTE-3 dataset for training and the Multi Layer Perceptron as the training algorithm.

---

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

The initial approach proposed in (Castillo, 2010a) is further explored in (Castillo, 2010b), which describes the system used for the RTE-5 evaluation campaign. It uses a supervised machine learning approach to train a SVM classifier over a variety of lexical and semantic metrics (most commonly, distances and similarities). The training is carried out over combinations of the RTE-3, RTE-4 and RTE-5 training sets, and the results obtained are slightly above the median for both the two and three way tasks.

(Castillo 2011) further attempts to improve the previously described system by training a Support Vector Machine classifier using features derived from lexical-semantic similarity metrics at the word and sentence level, and co-reference analysis as a means of pre-processing. The various semantic similarity measures used as features in the process of training the SVM engine are computed using a series of metrics described in literature on the basis of the WordNet taxonomy. Training was carried out over the RTE-3 dataset and the RTE-3-4C and RTE-4-4C datasets[13]. The reported results are well below the median.

The system described in (Castillo, 2010b) was also used for solving textual entailment in Spanish, as described in (Castillo 2010c). The features and training method were the same as those described previously, with the major difference represented by the language taken into account. For training, the author used a Spanish dataset, the SPARTE Corpus, which contains approximately 3000 labelled entailment pairs, and translated versions of the RTE-3, RTE-4 and RTE-5 datasets. The results obtained are promising, but the syntactic simplicity of the SPARTE corpus somewhat reduces the performance of the system.

(Montejo-Raez et al., 2011) describes an approach to solving the textual entailment problem in the summarization setting (RTE-6), and is interesting because of the machine learning method employed, which consists of combining the results of two SVM learners into a Naive Bayes learner. The first SVM is trained on the basis of Personalized Page Rank vectors (PPVs consists of a ranked sequence of WordNet synsets, weighted according to a random walk algorithm, inspired originally by the Google PageRank algorithm). The second SVM is trained using three features that are Named Entities matching counts for the text and hypothesis (the number of common NEs, for example). The results obtained are low, below the median, and this

[13] The -4C extensions are available at http://www.investigacion.frc.utn.edu.ar/mslabs/~jcastillo/Sagantest-suite/

is mainly due to the simplicity of the system (particularly when considering named entities). However, the use of stacking for merging different features results in a relevant increment in performance, which makes this approach noteworthy.

(Malakasiotis, 2010, RTE5) describes a machine learning approach to solving textual entailment that uses a Maximum Entropy (ME) classifier to solve the tree-way textual entailment challenge. The system uses a series of string similarity measures as features for determining entailment; because of the significant difference in length between the text and the hypothesis, most of the standard similarity measures are actually computed between the hypothesis and parts of the text. Syntactic and semantic features are also used to a small extent, in the form of the synonymy relation in WordNet and a series of measures derived from the syntactic parsing (text dependency recall, text dependency precision and their F-measure). The system result for the RTE-5 test set is very close to the median for both the two and three-way tasks.

Another Maximum Entropy based approach is described in (Ren et al., 2010). It makes use of lexical features (Jaccard distance, Bag of Words, Longest common substring, Levenstein distance), syntactic based features (Bag-of-Dependencies, Descendent Relation Features, Combined Verb Descendent Relations, Combined Subject Descendent Relations, Combined Subject-to-Verb Relations and Object Relations), lexical semantic similarity using WN and some special features extracted from Wikipedia for named entities. The results obtained are reasonably good, slightly below the median.

One of the more successful machine learning based approaches is that of (Ferrandez et al., 2010), which is based on the Weka's Support Vector Machine algorithm implementation, and uses a large number of features. The features vary significantly in type: various String-based Similarities (Levenshtein distance, Jaro distance, cosine similarity, etc.); WordNet-based similarities, weighed with IDF scores (such as the Lin similarity measure); antonymy, based on WordNet for nouns and VerbOcean for verbs; the number of named entities form H that appear in T; verb relations on the basis of WN (synonymy), VerbNet (in the same class) and VerbOcean (connection), and FrameNet for frame elements overlapping that shows how many frame elements from the frames detected in both T and H share similar or lexically related instantiations. The complexity of the implementation, particularly the use of many semantic features, means that the results obtained over the RTE-5 test set are good, well above the median for both the two and three-way tasks.

Another successful machine learning based systems is that of (Wang et al. 2010). It is based on the idea of Textual Relatedness measurement. Textual relatedness is based on the idea that, in order for the text to entail or contradict the hypothesis, it has to be semantically related to the hypothesis. If this is not the case, no significant relation can be drawn between the text and the hypothesis, and, therefore, the result for the entailment pair is Unknown. The system breaks down the three-way classification into a two-stage binary classification and focuses on the first stage as a subtask of the main task, which is to determine whether H is related to T. The most related sentences from T and H are extracted on the basis of a joint syntactic-semantic representation, and the most related sentence pairs from T and H are divided into dependency triples (i.e. <predicate, relation, argument> for semantic dependencies and <parent, relation, child> for syntactic dependencies). Various high-precision modules assign them scores on the basis of lexical matching, performed using WordNet and VerbOcean. A voting system is then used in order to make the final decision, and the system also includes some fallback strategies, for those cases where the main, high precision strategies fail to give a result. The system scored good result in RTE-5, and was below the top scoring system by less than 5%.

(Volokh et al., 2011) is an adaptation of the system described in (Wang et al., 2010) for the RTE-6 challenge. The new system has a single component (no main and fallback strategies as before), which incorporates all knowledge sources needed for determining entailment. This design allows for high robustness and allows for the adding and removing of knowledge sources in order to measure its contribution. Since the system only uses one model, the voting mechanism is no longer a factor. Broadly, the features used for predicting entailment are based on the dependency structure similarity, the word similarity and the named entity similarity between the text and the hypothesis. In order to compute the dependency structure similarity, the system transforms the parse tree of a sentence to a "triple representation", which is a triple of the form <parent, label, child>. Based on the triple representation, NE recognition, and lexical similarity, a large set of features is defined, and then used for training a linear classification engine. The reported results are good, above the median for the RTE-6 main task. Considering the results obtained for both the RTE-5 and the RTE-6 tasks, we are confident in saying that the system is one of the best machine learning based approaches in literature.

As we have shown in this section, machine learning techniques are used extensively for solving textual entailment. This, coupled with the high results obtained over RTE datasets (many machine learning systems scored above the median in RTE-5 and RTE-6) and high potential for adaptability for different datasets and languages means that this type of approach is one of the most effective employed for solving textual entailment.

2.7. Entailment Rules

One of the most intuitive and successful approaches to solving textual entailment is that of using rules or heuristics. The rules employed vary widely from system to system, both in terms of the particular rules used and in terms of the variables of the rules.

One of the best rule based systems was the winning entry for the RTE-4 competition three-way task, the one described in (Iftene, 2008) and (Iftene, 2009). Additionally, this system came in second for the two way task of RTE-4. The addition of supplementary semantic tools improved the basic system even more, so that it scored highest in both the two and three way task of RTE-5 (Iftene and Moruz, 2010).

This textual entailment recognition system was initially introduced by (Iftene and Balahur-Dobrescu, 2007), and further developed in (Iftene, 2008), (Iftene, 2009), (Iftene and Moruz, 2010), (Iftene and Moruz, 2011) and works by mapping every entity from the hypothesis to a node from the text, using extensive semantic resources such as DIRT, Wikipedia, WordNet, VerbOcean, VerbNet and an acronym database. The result of the mapping process is a fitness value associated to every word in the hypothesis; the local fitness values are then used to compute a global fitness value, which is then used to classify the current entailment pair as positive, contradiction or unknown entailment. The positive, contradiction and unknown clusters are defined as disjoint intervals between -1 and 1, according to data determined from training sets.

The pre-processing required for running the entailment system is extensive, and includes parsing of both the text and the hypothesis for dependency trees with MINIPAR (Lin, 1998), correcting of the POS (Part Of Speech) tagging with TreeTagger (http://www.cele.nottingham.ac.uk/~ccztk/treetagger.php) and determining named entities, using LingPipe (http://www.alias-i.com/lingpipe/) and GATE (Cunningham et al., 2001). Further preprocessing determines the types of numerals (whether they are percentages, measure values, time periods, order numerals) (Iftene, 2008).

The objective of the system described in (Iftene, 2009) is to map every node in the hypothesis to a node in the text, and, in the process, to calculate a local fitness value for the mapping. The author distinguishes between two possible cases: direct mapping, where the entities in the hypothesis are found in the text, and indirect mapping, in which case the entities from the hypothesis need to undergo certain transformations in order to be successfully mapped to entities in the text. According to the quality of the mapping, a local fitness score is awarded to each entity, and the local scores are then combined and normalized in order to obtain a global score, which will solve the entailment. Indirect mapping is carried out using a series of rules, specifically written for each of the possible solutions.

In the case of positive rules, any sort of mapping will increase the global score by a given amount, as the basic idea is to find a way of transforming the hypothesis into a form which is, more or less, similar to the text. For example, the DIRT resource considers the verbs "claim" and "declare" similar, and, therefore, the sentence "Obama claims victory" is transformed into "Obama declares victory". Also, in the case of named entities, the acronym "EU", for example, will be mapped to "European Union", the word "Ireland" will be correlated with "Irish", etc. Also, a number of rules are used for transforming natural language numerical values into intervals or numbers, such as "at least 80%" which entails "at least 70%" and "killing 109 passengers on board and 4 ground crew" which is equivalent to "killing 113 people" (Iftene, 2008), (Iftene, 2009).

Negation rules firstly penalize nodes in the hypothesis that could not be mapped to the text. Next, the verbs that could be matched are checked for negation, whether it be antonymy or negated synonymy. Antonymy also concerns contradiction between numerical values, for example. Also, words such as "different", "distinct", "separate", etc, are considered cues for non-entailment.

Unknown cases are cued by the use of words such as "may", "maybe", "possible", "rumour", etc, and the system introduces minor penalties whenever it encounters them. Also, the apparition of a new named entity in the hypothesis (by new the author understands a named entity that does not appear in the text) sets the final result for the pair at unknown. In the case of numeric values, the unit of measure is also kept, as differences most often lead to unknown cases.

In 2009, the system described above was further improved, mainly with the addition of more detailed semantic knowledge and with more refined pre-processing, as described in (Iftene and Moruz, 2010). The system performed very well, with an accuracy of 73.5% for the two way task (5% higher than the second best scoring system) and an accuracy 68.33% on the three way task (again 5% better than the second best scoring system). Compared to previous approaches, the largest increase in performance came from better use of named entities and from improvements in performance, as shown by ablation tests. In order to compensate for possible misspelling of named entities, the system also employs the Google API. The system also performed well when applied to the pilot task, coming in fourth out of 8 participants, with an f-measure that was very close to the median (31.61%). The lesser performance for the pilot task is mainly due to the fact that, because of time constraints, only a lightweight version of the system could be applied to the test data.

The UAIC Participation at RTE-6, described in (Iftene and Moruz, 2011) is an adaptation of the RTE-5 participation to the newly introduced main task in a summarization setting. The main improvement to the system was the addition of VerbNet as a resource for verb matching. Smaller adaptations were also carried out, mostly taking into account the differences brought about by the changing of the type of data sets. Another significant change was the use of three sets of thresholds for deciding entailment, as opposed to just one in previous versions, which allowed for varying the system's precision and recall. Also, since the system was originally developed for the three way entailment task, it could easily be adapted to the two way entailment task. The variance in precision and recall was needed for adapting to the novelty detection subtask. The results obtained for the main task were good, falling slightly below the median, with the best result coming from the configuration of the entailment thresholds that was designed to balance the values of recall and precision (22.89% precision, 27.20% recall and 24.85% f-measure). The results (below the median) are due to the fact that the summarization based entailment task has significantly different data sets, as compared to previous RTE challenges, to which the system was not sufficiently adapted. Also, the system did not make use of any discourse derived information, such as coreference resolution, for example, which further lowered the results. For the novelty detection subtask, however, the results were very good, with one of the runs (the run favouring recall over precision) scoring fourth overall, 4% below the top system. The results obtained for the novelty detection subtask prove that the system works well, the lower results on the main task being mainly due to certain incompatibility between the system's initial setting and the problem at hand, a fact also shown by the ablation tests.

A more recent heuristics based approach to solving textual entailment is Sangyan, described in (Shivhare et al., 2011). The authors try to solve entailment by using a two step strategy: they attempt a syntactic match between the text and the hypothesis, and, in the cases where this does not confirm entailment, they attempt to detect the semantic relatedness between the text and the hypothesis. To determine syntactic similarity between text-hypothesis pair, Sangyan uses two sub-modules: dependency tree match module and a heuristics module. The dependency tree match module attempts to match subtrees from both the text and the hypothesis, with the aid of some lexical-semantic tools, such as coreference resolution, named entity recognition or lemmatization. The system also makes use of resources such as WordNet and Verb Ocean for matching the different entities. Since syntactic matching techniques cannot map semantically similar but syntactically different constructs, the heuristics module attempts to transform different syntactic relations to similar representations, such as relations induced by copulative verbs and appositions. In order to handle syntactic variability, instead of a typical rule based approach, Sangyan attempts to perform semantic matching by using FrameNet frames and the Shalmanesar semantic parser for „Semantic Clustering" of words. The semantic match between text and hypothesis is defined in terms of frame overlap between the text and the hypothesis, weighed down with the lexical match coefficient. This approach is hindered, however, by two important drawbacks: the limited number of heuristics used to perform syntactic matching and the limited coverage of FrameNet frames, which contain approximately 6500 annotated predicates, insufficient to train real world applications. These drawbacks may be the reason for the middle scale performance of this system in the RTE-6 main challenge, where it scored an accuracy of 29.64%.

The Bar-Ilan University Textual Entailment Engine – *BiuTee* – (Mirkin et al., 2010), (Stern et al., 2011) is a transformation-based entailment system making use of various types of entailment knowledge. The knowledge is represented uniformly in the system in the form of entailment rules (which are generally represented as LHS => RHS), which permits the system to apply the same kinds of transformations to the text regardless of the source of the knowledge. By applying the transformation rules to the text, the system generates a set of consequents, which are stored in the form of parse trees in a packed representation, named Compact Forest (Bar-Haim et al., 2009). The basic idea employed by the system is to gradually transform the text, in order to make it more similar to the hypothesis, while ensuring that entailment holds between the original text and its consequents. As a final step, the entailment decision is taken depending on the result of an approximate syntactic match between the final consequent and the hypothesis, in order to compensate for possible knowledge gaps in the rules.

The 2010 version of the BiuTee system, which was used for the RTE-6 main task, underwent major improvements: the approximate matching component was replaced by a syntax-based matching algorithm and the lexical entailment rules were chained via a novel entailment graph algorithm. The compact forest was obtain by applying three types of rules: a set of human created generic syntactic rules, a set of rules obtained as described in (Aharon et al. 2010), and an extension of these rules on the basis of the entailment graph algorithm. The syntax based matching algorithm is based on the assumption that the absence of some syntactic dependencies means that the hypothesis cannot be entailed by the text, while the absence of others is not relevant. Therefore, the algorithm consists of matching verbs from the text and the hypothesis and then matching their *subject* and *object* dependants in a bag of words manner; if the number of different dependencies is below a certain threshold, or if the bags of words are covered to more than a given threshold, the algorithm returns entailment. The system performed quite well on the RTE-6 test data, with an f-measure of 37.50%, well above the median, making the 2011 version of BiuTee one of the best systems developed for solving the task of textual entailment in a summarization setting.

The compact forest described in (Bar-Haim et al. 2009) is an attempt at decreasing the number of derivable sentences obtained by applying transformation rules to a text. A compact forest is a novel data structure proposed by the authors, which represents a large set of trees in a compact manner, for reasons of complexity. When using this structure, each rule application generates only those nodes on the right-hand-side of the rule, while the rest of the consequent tree is shared with the source; in addition to the reduction in required space, this type of representation also reduces the number of redundant rule applications. The representation of a compact forest is based on the notion of disjunction edges, an extension of dependency edges that specify a set of alternative edges of multiple trees. The authors also describe an algorithm for determining inference over compact forests. Testing of the method showed that it significantly improves running times, reduces the number of rule applications and reduces the number of generated nodes, proving the value of the approach.

(Krestel et al., 2010) proposes the idea of an artificial believer, which is designed to acquire knowledge by processing texts and to maintain a consistent belief base by detecting inconsistencies and rejecting information according to a belief revision strategy. In order to construct and maintain the belief base, the system employs a Fuzzy Believer that is intended to detect conflicting information; the task solved by the Fuzzy Believer is isomorphic to recognizing textual entailment. Entailment is computed over the predicate-argument structures (PAS), typically triples in the form of (subject, predicate, object), extracted from the text and the hypothesis. The system uses heuristics to compare these predicate-argument structures, and if the similarity score of the merged structures is above a threshold, the result is entailment. The results obtained using this method are middle range, very close to the median of the RTE-5 main task, but the value of the system lies in the manner in which it creates the belief system for detecting entailment, which is very well suited for the task of solving textual entailment in a summarization setting, such as the RTE-5 pilot task.

Many rule based systems make use of automatically extracted rules, and as such, the problem of automatically extracting rules becomes relevant for the field of textual entailment. Even though most authors focus on unsupervised acquisition of entailment rules (rules that apply on templates with two variables) and ignores unary rules (rules that apply on templates with a single variable), there is a necessity for such rules. (Szpektor and Dagan, 2008) investigate two approaches for unsupervised learning of unary entailment rules and subsequently compares them with a binary rule learning method. The main approach consists of learning rules by measuring the similarity between the variable instantiations of two templates in a corpus. Learning is performed by adapting state-of-the-art similarity measures for unary rule learning, but the authors also propose new measure, named Balanced-Inclusion, which uses the notion of symmetric semantic similarity and adapts it to the directionality needed in entailment. Balanced-Inclusions identifies entailing templates on the basis of a directional measure and penalizes infrequent templates using a symmetric measure The second approach consists of deriving unary rules from binary rules learned by state-of-the art binary rule learning methods. After comparing the method to state of the art binary rule extraction systems, the authors have determined that unary rules perform better, and that learning unary rules from a corpus yields better results than extracting them from binary rules.

(Aharon et al. 2010) explores how FrameNet (Baker et al., 1998) could be effectively used for generating entailment rules between predicates. In the general case, in order to generate entailment rules, one need to address two issues: a) identifying the lexical entailment relations between predicates, e.g. 'cure → recover'; b) mapping argument positions, e.g. 'cure X → X recover'. The main approach for generating highly accurate rule-sets is to use manually constructed resources, with WordNet being the most popular one. WordNet, however, is limited for the task of extracting entailment rules, mainly because many relations, such as "cause", are not fully mapped (e.g. "elect→vote"), and many of the relations that are actually present do not preserve argument position (such as the relation between 'kill → die'). FrameNet is more suited to the task of automatically extracting entailment rules because it has more relations suited to the task of solving textual entailment and every frame has explicitly stated arguments. The authors describe an algorithm for extracting entailment rules from FramNet, which works in three steps: a) extracting templates for each lexical unit (LU); b) detecting lexical entailment relations between pairs of LUs; c) generating entailment rules by mapping the arguments between two LUs in each entailing pair.

For each sentence of any given LU, the algorithm extracts unary templates, by decomposing templates with multiple arguments into templates with only one argument. Templates are obtained by first replacing the frame element phrases with the frame element names, and then generating a rule by extracting a path in the parse tree between the LU and the FE. Relations between LUs are extracted according to a series of heuristics (morphological derivations are marked as paraphrases, dominant LUs are entailed by non-dominant LUs in the same frame, relations between frames, etc.). Lastly, the algorithm detects common frame elements for each identified relation between LUs and generates all possible rules from that relation. The authors have manually checked 250 entailment rules obtained by the algorithm, and have judged approximately 69% of them as correct, a significant improvement over previous rule extraction systems.

(Berant et al., 2010) discusses a wider view of the issue of learning entailment rules. Most previous work focuses on the learning of entailment rules in an isolated manner, but the authors propose that entailment rules interact with each other (for example, some entailment rules are in a transitive relation), and they take advantage of this fact for improving entailment rule learning. The authors define a new type of data structure, which they call an Entailment Graph, which has propositional templates for nodes and entailment relations for edges. A propositional template is defined as a path in a dependency tree from one argument to another, so every template contains a predicate; each sense of a polysemantic predicate must correspond to a separate template. Since the entailment relation is transitive, then the entailment graph itself is transitive, and entailment paths can be created between two nodes. The authors then present an algorithm for learning the edges of an entailment graph given its set of nodes. It begins by preprocessing of the node, using a large corpus and WordNet for training an entailment classifier that estimates the likelihood that one propositional template entails another. Given the graph nodes and the estimated entailment rules detected during preprocessing, a global optimization approach is employed for determining the set of edges that maximizes the probability (or score) of the entire graph, given the edge probabilities (or scores) supplied by the entailment classifier and the graph constraints (transitivity and others). The optimization problem is formulated as an Integer Linear Programming problem. The results clearly demonstrate that a global approach improves performance on the entailment graph learning task, and the overall advantage of employing an ILP solver rather than a greedy algorithm.

(Dinu and Wang, 2009), on the other hand, explores ways of improving existing inference rule collections. The approach is based on two observations: some rules required for solving entailment are not present even in the largest of collections, such as DIRT and some systematic errors in rule collections can be excluded. The problem of missing rules is addressed by combining DIRT with a hand annotated lexical resource in order to expand the existing rule set. The authors extended the DIRT rule set by expanding existing rules with synonyms from WordNet (all the lexical items involved in rules are replaced with each of their synonyms in any possible combination allowed by the synsets). The procedure is successful in adding missing rules, but, because of the lack of word sense disambiguation prior to the expansion, many incorrect or nonsensical rules are also produced. This, however, is not relevant for the current approach, as the purpose of the authors is to evaluate the extended rule set in an application setting and not as an independent collection of entailment rules. The authors also remove some of the most systematic errors in the DIRT rule database. The errors are due to the nature of the

Distributional Hypothesis algorithm; DH algorithms do not only extract phrases that have similar meanings, but also phrases that have opposite meanings. To compensate for this problem, rules that contain WordNet antonyms are eliminated from the rule set. The coverage of the new, extended, rule set is evaluated, and it is found low, but the improvement in precision for the pairs to which the new rules apply is significant. Even though, on the whole, the improvement brought about by the extension of the DIRT rule set is not great, mostly because of the low coverage of the rules, this paper does prove the advantage of improving automatically extracted rules by using manually annotated resources.

2.8. Cross Lingual Textual Entailment

Even though the appeal of textual entailment increased steadily since its definition in (Dagan and Glickman 2004), a fact that has been also proven by the six editions of the Recognizing Textual Entailment Challenge, the main focus of the research in the field has been on TE for the English language, and extensions to the task, such as cross-lingual textual entailment, have been largely ignored. Despite this, the interest in cross-lingual applications for NLP does exist, and has grown steadily, a fact proven by the successful evaluation campaigns of the Cross Lingual Evaluation Forum (CLEF).

(Mehdad et al. 2010d) proposes the idea of cross-lingual textual entailment (CLTE) as a means of performing semantic inference across different languages. The authors adapt the definition of textual entailment for the cross-lingual approach as follows: cross-lingual textual entailment is a relation between two expanses of natural language texts in different languages, called text and hypothesis, and depicted as T and H, that holds if a human would infer that H is most likely true solely on the basis of the information in T, or, in other words, the meaning of H can be deduced from the meaning of T. In order to solve the problem of CLTE, the authors propose two methods: either bringing the cross-lingual textual entailment decision in the monolingual case by translating the text, the hypothesis or both, or embedding cross-lingual processing techniques in the textual entailment engine, thus circumventing the need for translation.

The simplest method for solving cross-lingual textual entailment is to attach a machine translation system as a front end to a monolingual textual entailment engine. Considering the case of an English text and a French hypothesis, the purpose of the MT component is to translate the hypothesis to English, and then run a TE system on T and the translation of H. The advantage of this approach is that it is modular, which allows for easier development, debugging and maintaining of the system. Also, a modular system can be adapted to any language pair by attaching the appropriate translation systems; the approach can be further refined by using English as a pivot language, thus allowing for the use of the same TE system for solving CLTE on any language pair, given a translation module is available for each language. The main drawback of this approach is the performance of state of the art machine translation systems, which may introduce and propagate errors. To compensate for this, the authors propose passing the top best translations through the entailment system, and then choosing the appropriate result on some combination of criteria.

The more advanced method for solving CLTE is to integrate MT and TE techniques, thus creating a system capable of solving the cross-lingual textual entailment problem without the need for translation, and with lower complexity. A possible approach is to extract information from phrase-tables used by phrase-based machine translation and enrich entailment rules (for example, the relation between the French phrase “ordinateur portable” and the English phrase “laptop”).

A study of the feasibility of the basic method of CLTE was carried out. The corpus used was the RTE-3 development set, with English texts and French hypotheses (the French translations were manually obtained). The hypotheses were automatically translated into English by using the Google translator and another state of the art machine translation system, Moses. The entailment system used is a version of the EDITS system (Kouylekov and Magnin, 2006), trained on the RTE-3 test set. The results show that using Google to translate the French hypotheses into English yields the same result as that obtained by running the system on the original RTE-3 development set.

2.9. Discourse knowledge

With the introduction of the RTE-5 pilot task, which then became the main task of RTE-6 and will be the main task of RTE-7, the necessity for detecting discourse references, such as corefence and bridging anaphora has become more apparent. This section describes some of the approaches to solving textual entailment by using such discourse knowledge.

(Litkowski, 2010) describes a system that uses only routines to examine the overlap of discourse entities between the text and hypothesis. Discourse entities (represented at the text level by noun phrases in most cases) are extracted by means of the CL Reasearch Linguistic Task Analizer and are accompanied by a set of attributes such as WordNet sense number, anaphoric references to discourse entities earlier in the text, number, and type. Broadly, the algorithm attempts to match discourse entities from the text to the hypothesis on the basis of the attributes attached to them. If at least half of the discourse entities in the hypothesis are matched to discourse entities in the text a second routine, that checks for verb and subject consistency. On the basis of this match, an entailment decision is picked. The system described in this paper was tested on the RTE-5 data sets; the results obtained were middle range, very close to the median. The results prove the value of the approach, but the system is hindered because much of the information generated at the preprocessing stage is not used in the entailment decision.

(Mirkin et al., 2010a) analyzes the roles of discourse references for textual entailment inference, as a means of highlighting promising directions for the incorporation of discourse phenomena into inference. An important consideration is that the authors only wish to examine the effect of discourse references on the process of entailment and, as such, manually annotate only those references that are relevant to solving TE on the RTE-5 search data set. These choices are supported by the fact that they solve a number of problems faced by previous researchers that attempted to integrate discourse references in the entailment process: discourse phenomena are well represented in the RTE-5 search sets, the references are manually annotated and cover all syntactic categories, rather than just nominal coreference. Since only the discourse references that are relevant to the entailment process are annotated, the authors define three elements that make up the reference: the *target component* in H (the subtree in H that cannot be supported by local information in T), the *focus term* in T (the expression in T that does not cover the target component by itself, but participates in a reference relation that can cover it), and the *reference term* which stands in a reference relation to the focus term.

In the author’s opinion, the integration of discourse references in previous approaches was overly simple, and thus contributed to the low improvement brought by the component. To address this issue, and on the basis of the manually annotated corpus described above, they propose three distinct cases for integration, which correspond to different relations between the target components, the focus term and the reference term: substitution (where the focus term is replaced with the reference term), merge (in cases where a match for the entire target component cannot be found in the text and is scattered among multiple locations) and insertion (in the case of bridging references).

The analysis carried out by the authors shows that in 44% of the entailment pairs examined, the correct result could not be extracted without the use of discourse references. They have also shown that, although the large majority of reference cases are nominal, there also are a significant number of cases of verbal references. Finally, this study shows that textual entailment can profit substantially from better discourse handling; it also demonstrates that improvements are required for both the discourse reference resolution systems, so that they can cover a larger number of reference types, and the integration of the discourse information supplied by the reference systems, on both the new types of references and even for those relations which are already covered by existing resolvers.

(Mirkin et al, 2010b) also discusses the use of discourse information for solving textual entailment. The authors start off with a state of the art textual entailment engine, BiuTee (Bar-Haim et al., 2009) which was enhanced by addressing prominent discourse phenomena, significant for entailment, discussed below.

To compensate for coreference relations that are not identified by off-the-shelf resolvers, the authors define the notion of augmented coreference set. This is based on the observation that a large number of coreference relations share some of their lexical elements, and most of these are missed by resolvers. To identify these additional coreferences, the authors consider that two noun phrases in the same document are corefering if their heads are identical and no semantic incompatibility exists between their modifiers (no mismatching numbers, no antonymy and no co-hyponymy). The system also makes use of global information, in the sense that some key pieces of information, such as the document title or information given in the first few sentences, are considered known for the whole document. Key words are extracted by means of tf-idf scores, and then added to each candidate sentence to compensate for the phenomenon of bridging. Information from the sentence preceding the candidate sentence is also extracted, as another way to address the problem of missing coreference and bridging. Based on the observation that entailing sentences are usually clustered, the authors also employ a meta-classifier that uses the output of another classifier as input in order to take the final entailment decision. The meta-classifier makes use of some other meta-features, such as title entailment (whether the title or the first sentence of the document entail the hypothesis), second-closest entailment, smoothed entailment (smoothes the classification of a sentence with the scores of neighbouring sentences) and first sentence entailment (whether the first sentence of the document entails the title). Test carried out with the BiuTee system enhanced as described above proves that the use of discourse information does indeed improve the performance of textual entailment in a summarization setting.

2.10. Uses of Textual Entailment

When it was first defined in (Dagan and Glickman, 2004), textual entailment was described as a means of reducing natural language variability (semantically similar texts that are significantly different at the lexical and syntactic levels). Since the problem of language variability greatly reduces the performance of many NLP applications, it stands to reason that the use of a state-of-the-art textual entailment engine brings a significant improvement. This section provides a brief overview of some of the most recent uses of textual entailment in the field of Natural Language Processing.

A common problem faced by machine translation system is the need to translate terms that have no correspondence in its lexicon or paraphrase database. (Mirkin et al., 2009) proposes a novel approach that uses paraphrase rules together with textual entailment rules to fill in the knowledge gaps in order to improve the precision of machine translation. Previous approaches to translating unknown terms (unknown in the sense that they have no translation equivalent) typically extend the phrase table of the statistical machine translation system with paraphrases extracted from multilingual corpora. The idea is to use monolingual paraphrasing methods and resources in order to obtain a larger set of rules for paraphrasing the source instead of the translation. The rules are then applied to the source sentence before the translation step, thus eliminating the problem of lack of knowledge at the moment of translation. The rule set can be expanded even more, by means of directional entailment rules in those cases where paraphrases cannot be used, that would generate more general versions of the source text. This approach, based on the framework of textual entailment, considers the newly generated texts (by means of asymmetric rules) as being entailed by the original one. An algorithm for the transformation of the source sentence into a translatable version is proposed; roughly, it first attempts to find paraphrases for unknown terms, and, in case that fails, it attempts to apply asymmetric entailment rules. The score for each rule application is computed, in order to rank the set of candidates obtained through these transformations, and the algorithm can then return either the translation of the best scoring candidate or the top *n* candidates, which are translated and then ranked using target language information. The approach was tested by training a SMT system on a corpus consisting of approximately one million English-French sentence pairs from the Europarl corpus, which was then used to translate a set of almost 6000 from English to French. The results were manually evaluated by native speakers, and it was found that, even though the application of entailment rules did slightly reduce precision, coverage increased dramatically (precision is the amount of correctly translated terms divided by the total number of translated terms, and coverage is the total number of translations divided by the number of terms in the source language).

(Pado et al., 2009) discusses the use of textual entailment as a means of automatically evaluating statistical machine translation. The fundamental problem in the automatic evaluation of SMT is that it is difficult to account for linguistic variability between the human translation and the machine translation. State of the art measures attempt to compensate for this fact by computing n-gram overlap (BLEU, NIST), by making the matching process more intelligent (TER), or by using linguistic evidence such as lexical similarity (METEOR); these metrics, however, usually concentrate on one type of linguistic phenomenon, which does not correlate well with human annotation. The authors explore two main ideas: a combination of all the classic metrics into a more robust measure or the use of semantic equivalence by means of symmetric textual entailment. The textual entailment engine used was that of the Stanford Entailment Recognizer. Experiments were carried out over a well known corpus, and proved that evaluating statistical machine translation using textual entailment yields better results than using current state-of-the-art metrics; however, the combination of classic metrics and textual entailment performs best, which suggests that each approach addresses somewhat different linguistic phenomena.

(Agerri, 2008) proposes to address the problem of understanding the figurative use of language, particularly metaphors, by adapting the Recognizing Textual Entailment (RTE) framework for metaphor interpretation. By analyzing results reported over the RTE-1 and RTE-2 datasets, the author draws two significant conclusions: the ability of processing metaphors may improve the performance of textual inference systems and the RTE setting could provide a general semantic framework for evaluating and testing theories that attempt to explain the usage of metaphor in text.

(Sacaleanu, 2008) describes a Question Answering system which retrieves answers from structured data regarding cinemas and movies using a multilingual ontology and textual entailment. The QA problem can be treated as an entailment problem, by assuming that the question itself is the text (in its affirmative version) and the hypothesis is a relational answer pattern, associated to instructions for retrieving the answer to the input question. Therefore, the problem of question answering is reduced to determining entailment between the question and a set of relational answer patterns. The idea is applicable because of the relatively low coverage of the system, which permits only a small number of types of questions, and, therefore, a small number of search patterns that need to be tested for entailment. Since the database is in the form of a multilingual ontology, the only language dependant component is the entailment engine, which means that the system can be easily adapted to new languages. The results obtained were promising, but vary widely from language to language, mostly because of the quality of the entailment engine used.

(Wang and Callison-Burch, 2010) describes an experiment aimed at creating a set of more natural hypotheses for a set of texts by using Amazon's Mechanical Turk to generate facts and counter-facts from texts for certain named entities. This experiment is prompted by questions regarding the relevance of artificially created hypotheses, particularly in the case of negative examples. Human annotators receive a paragraph of text and a highlighted named-entity; their task is to compose facts and counter-facts about the named-entity in that context. The analysis of the results is performed by comparing the acquired data against the RTE-5 dataset, and the findings show that, on average, facts have two more words than the hypotheses provided by the organizers, and counter-facts are significantly longer then negative hypotheses (5 words on average).

(Sammons et al., 2010) performs a very insightful analysis over the results reported by the systems participating in the RTE-5 evaluation campaign, and suggest a series of potential improvement points on the basis of common errors observed. The system analysis consists of a series of questions whose answer provide potential directions for improvement. The intuitive first question is “What does a system need to do to improve its performance?”; according to the reported results, among the most common causes for error among top systems were failure to perform numeric reasoning or failure to match named entities or numerical quantities. It is also shown that there are a number of “hard examples” (pairs on which at least 4 of the 5 top systems chose the incorrect solution), which are strongly correlated to deeper lexical relations between words and the need for external knowledge sources. The authors also show that accurate identification of the linguistic phenomenon involved in the relation between text and hypothesis significantly increases the accuracy of textual entailment systems.

3. Solving Textual Entailment by Semantic Means

The notion of textual entailment was first described by (Dagan and Glickman, 2004) as an attempt to generalize the issue of natural language variability across NLP tasks, such as question answering, automatic summarization or information extraction. According to (Dagan and Glickman, 2004), "textual entailment (entailment, in short) is defined as a relationship between a coherent text, **T**, and a language expression, which is considered as a hypothesis, **H**. We say that **T** entails **H** (**H** is a consequent of **T**), denoted by **T⇒H**, if the meaning of **H**, as interpreted in the context of **T**, can be inferred from the meaning of **T**." The entailment relation defined above is directional since, while the meaning of one text may imply the meaning of another, the opposite does not always hold true.

The definition given above, while complete and correct, it is too abstract to be directly used in practical applications of Natural Language Processing. The Recognising Textual Entailment Challenge (RTE) is the main series of evaluation campaigns for textual entailment engines, which was initially put forth in 2005 by PASCAL (Pattern Analysis, Statistical Modelling and Computational Learning – http://www.pascal-network.org/) – the European Commission's IST-funded Network of Excellence. Therefore, there have been a number of adaptations of the initial definition for textual entailment, in order to make it more practical for the purpose of real life implementations.

In the course of analysis of various RTE datasets, we have come up with the intuition that entailment pairs can be solved, in the majority of cases, by examining two types of information, which lead to a semantic understanding of the text and the hypothesis:

- The relation of the verbs in the hypothesis to the ones in the text. In this context, verbs are taken to mean the verb itself, together with its arguments and adjuncts; thus, the comparison of two verbs is a comparison of complex structures, which are, in essence, atomic propositions (clauses). If the verbs, together with their arguments and adjuncts, match over **T** and **H**, we have **ENTAILMENT** (by matching we understand that $\forall\ verb\ q\ \in \boldsymbol{H}, \exists\ verb\ p \in \boldsymbol{T}\ so\ that\ p \rightarrow q$; we say that a predicate p entails a predicate q, p→q, if q is a consequence of p, or p and q are synonyms, or q is a subevent of p).

- Each argument or adjunct is an entity, with a set of defined properties. It may happen that, despite agreement at the level of the verb, there may be differences in terms of entity properties for similar arguments or adjuncts. In order to solve such cases correctly, each argument and adjunct is considered an entity with an attached set of attributes, and only if they match we have **ENTAILMENT**. By matching at the argument level we understand that, given the feature sets for the arguments arg_T and arg_H, the unification of these feature structures (as defined in unification grammars) is successful and is equal to the feature structure of arg_T.

The intuition given above is based on the idea that if a text T entails a hypothesis H, it can be stated that H can be deduced from T, $T \vDash H$. Since natural language sentences are generated by predicates, the previous relation can be decomposed into $\forall\, Q(arg_1, arg_2, \dots arg_n)\ predicate\ in\ H,\ \exists P(arg_1^{'}, arg_2^{'}, \dots arg_m^{'})\ predicate\ in\ T$ so that $P(arg_1^{'}, arg_2^{'}, \dots arg_m^{'}) \vDash Q(arg_1, arg_2, \dots arg_n)$. This formula can be further decomposed into $P \Rightarrow Q$ and $\forall arg_i\ \in \{arg_1, \dots arg_n\}, \exists arg_j^{'} \in \{arg_1^{'}, \dots arg_m^{'}\}$ so that $arg_j^{'} \Rightarrow arg_i$. The first part of the decomposition, namely the one referring to predicates is covered by the first part of the definition. The second part of the decomposition deals with predicate arguments, which are, in general, entities with property sets, and, therefore, the implication relation holds if and only if $arg_i \subseteq arg_j^{'}$, which is equivalent to a sucessful unification of the structures that describe the respective arguments, which is equal to the structure of the argument in the text. The reason we have chosen to use unification rather than the inclusion relation or the implication relation is that it is more intuitive and fits the available knowledge base structures (such as the WordNet taxonomy).

This section focuses on the relevance of verbs for textual entailment, as it is described in the next sections. In this respect we try to capitalize on our experience with the verbal group and predicate, which we have researched in the past. The results of this research are thoroughly described in (Curteanu et al. 2006a, 2006b, 2006c, 2007). The relation between argument property sets and entailment is also discussed. We also provide a general algorithm for solving Textual Entailment on the basis of the intuition given above, as well as a detailed algorithm for matching VerbNet descriptions of verbs automatically.

The idea of using verbs to solve textual entailment is not new, and has been used by (Burchardt and Frank, 2006) and (Hickl and Bensley, 2007), (Hickl, 2008) to name just a few. (Burchardt and Frank, 2006) approximate entailment in terms of structural and semantic overlap of text and hypothesis, combining wide-coverage LFG parsing with frame semantics, to project a lexical semantic representation with semantic roles, and then computes an overlap between **T** and **H**. (Hickl, 2008) extracts a set of commonly held beliefs, called discourse commitments, and once these are extracted, the task of solving entailment is reduced to determining whether the discourse commitments in **H** are supported by those in **T**.

The key difference between previous approaches and the solution described in this chapter is the fact that we try to exploit the notion of verbal alternations and Beth Levin's work on verb classes (also known as Levin classes, first described in (Levin, 1993)). To our knowledge, only one other textual entailment system has made use of the notion of Levin Classes and VerbNet, namely that of (Ferrandez et al., 2010); they do not make use of the full potential of the resource, however, and only check to see whether two verbs are in the same Levin class.

In more detail, the Levin class for each of the predicates in **H** is extracted, and then matched to predicates in **T** that have the same Levin class; if this fails, the verbs are matched on the basis of the semantic description attached to Levin classes. This procedure was tested by analyzing, in detail, 200 entailment pairs from the RTE-5 test set, and it was found that more than 38% of them can be solved by means of Levin class match. For identifying the Levin class of a verb we have used VerbNet, which is described in section 3.1.

Another novel feature of this approach is the grouping of entity properties into property sets; for two similar entities in T and H, entailment is defined as the result of the unification of these attribute sets. Out of the 200 analyzed pairs, over 29% can be solved using this method.

The rest of the chapter is structured as follows: section 3.1 is a description of VerbNet and the composition of classes in VerbNet; section 3.2 gives an algorithm for solving textual entailment on the basis of predicational semantics and argument structure matching, section 3.3 describes the analysis carried out over the set of 200 entailment pairs, and conclusions are given in section 3.4.

3.1. Levin's classes and VerbNet

The largest and most widely used classification of English verbs is that of Levin (1993), most commonly referred to as Levin classes. VerbNet (**VN**) (Kipper et al., 2000; Kipper-Schuler, 2005) is an online verb lexicon for English that provides detailed syntactic and semantic descriptions for Levin classes organized into a more refined taxonomy. According to (Kipper-Schuler et al., 2006), VN is a hierarchical, domain-independent, broad-coverage verb lexicon, and it has mappings to a number of widely used verb resources, such as FrameNet (see annex 4 for alignment example) (Baker et al., 1998) and WordNet (Fellbaum 1998).

A VerbNet class is completely described by a set of member verbs, the thematic roles for the predicate–argument structures of these verbs, selectional restrictions for the arguments and frames that contain a syntactic description and semantic predicates that also have a temporal function, in a manner similar to event decomposition (Moens and Steedman, 1988). VerbNet extends the original Levin classes by refining them into subclasses with a higher degree of syntactic and semantic coherence between members. It has been further extended with the addition of the resource described by (Korhonen and Briscoe, 2004), which includes 57 new classes and 106 new diathesis alternations. An example of a VerbNet class is given in annex 2.

3.1.1. Syntactic Frames in VN

Each VN class contains a set of syntactic descriptions (also known as frames) that represent potential surface realizations for the argument structure, such as transitive, intransitive, prepositional phrases, etc, and a number of diathesis alternations, given by Levin as part of each class (Kipper-Schuler, 2006). Each syntactic frame is described using thematic roles (Agent, Patient, etc), the verb, and any other lexical items needed for a construction or alternation. Thematic roles are sometimes subject to selectional restrictions (such as animate, organization, human, etc), and the syntactic frames themselves can be constrained with regards to what prepositions are allowed. Additionally, further restrictions can be imposed on the thematic roles, in order to give the syntactic nature of the constituent that is most likely to be associated with the thematic role. Levin classes are mostly characterized by **NP** and **PP** complements, although some classes, like *tell*-37.2, refer to sentential complements, while the classes described by (Korhonen and Briscoe, 2004) mainly focus on sentential arguments.

3.1.2. Semantic predicates

In addition to the syntactic description attached to each Levin class in VN, there is also a semantic description, which is described as a conjunction of Boolean semantic predicates such as "*cause*", "*motion*" or "*contact*". Each of these predicates is associated with an event variable **E** that is used for specifying the moment at which the predicate is true (*start*(**E**) for the stage before the action taking place, *during*(**E**) for the culmination stage and *end*(**E**) for the stage after the action taking place). Relations between verbs or classes of verbs, such as antonymy and entailment from WordNet, and relations between verbs or classes of verbs such as those found in FrameNet can be extracted by means of these semantic predicates. In VN, aspect is described by the event variable argument in predicates (Kipper-Schuler et al., 2006).

3.1.3. Statistics about VerbNet

At the moment, VN version 3.1, the latest version of the English VN, has descriptions for over 5800 verbs, distributed in 270 first level classes and 200 subclasses. The descriptions for these verbs use 33 thematic roles, 36 selectional restrictions, 296 primary frames (which can be extended by specifying the thematic role type or the complement type) and 145 semantic predicates. The lexicon also relies on a shallow hierarchy of prepositions, with 66 entries. VN coverage of PropBank tokens (Palmer et al., 2005) has increased in the current version to above 90%. Also, in VN 3.1, Levin's taxonomy has grown in both depth, with the refinement of several of the classes defined in (Levin, 1993) and in width, with the addition of the extension proposed by (Korhonen and Briscoe, 2004).

3.2. A Predication Based Algorithm for Soving Textual Enailement

The purpose of the feasibility study described in section 3.3 was to test whether predication driven textual entailment, based on Levin classes, is possible; the results have shown that not only is it possible, but that such an approach solves 38% of the entailment pairs taken into consideration; also, a further 29.5% of the pairs are solved by the use of argument structure matching. Because of this we propose an algorithm that solves verb based textual entailment, and that has also been extended to perform argument matching; analysis of the algorithm carried out over the 200 entailment pairs mentioned above showed that it can correctly solve entailment in most cases (difficult cases, which are not correctly solved by our algorithm, are discussed in section 3.4).

The algorithm receives as input syntactic trees for the text and the hypothesis (functional dependency parses) and returns as output the entailment value for that pair. When we make references to the sense of a word, we understand the most common sense of the word (in terms of WN frequency); also, when we refer to two words as being synonyms, we understand that there exists at least one synset in WordNet so that senses from both words belong to it. In terms of argument entailment, we say that an argument arg_T in T entails an argument arg_H in H if arg_T and arg_H are synonyms, they refer the same entity (by means of coreference or acrnym, for example), arg_H is a part of arg_T, arg_H is a hypernym of arg_T or arg_H is an antecedent of arg_T in the WN taxonomy. We say that arg_H is aligned to arg_T if arg_H is entailed by arg_T.

1. Extract the Levin class for all the verbs in T and H and attach the appropriate semantic description, on the basis of the Levin class and syntactic analysis;
2. For every verb *q* in H, determine whether at least one verb *p* in T has the same Levin as *q*
   a. For all candidates *p* in T, if the arguments and adjuncts match over *p* and *q*, and the verbs are not semantic opposites (e.g. antonyms or negations of one another), return ENTAILMENT
   b. Else,
      (i) if the arguments and adjuncts match, but the verbs are semantic opposites (e.g. antonyms or negations of one another), or the arguments are related but do not match return CONTRADICTION
      (ii) else if the arguments are not related, return UNKNOWN
   c. Else, return UNKNOWN
3. For every verb *q* in H, if there is no verb *p* in T has the same Levin as *q*, extract relations between *q* and *p* on the basis of Levin semantic descriptions
   a. If the verbs in H are not semantic opposites (e.g. antonyms or negations of one another)of verbs in T, and the arguments match, return ENTAILMENT

```
   b. Else,
      (i) if q is semantically opposite to p and the arguments
      match, or the arguments do not match, return CONTRADICTION
      (ii) else if the arguments are not related, return UNKNOWN
   c. Else, return UNKNOWN
4. Return UNKNOWN
```

The algorithm described above addresses the issues with regards to using Levin classes for calculating entailment. It starts off with attaching the appropriate Levin class for each verb in **T** and **H**. The next step is to attempt to match every verb in the hypothesis to at least one verb in the text on the basis of their Levin class value in VerbNet. Given the above verb matching, the algorithm then attempts to perform unification over the argument feature structures. If the arguments match, as defined above, then the result of the analysis is ENTAILMENT; in the case of argument contradiction, the result is CONTRADICTION. An example of argument contradiction is given below:

T: The cat ate a large mouse.
H: The cat ate a small mouse.

If the verb in the hypothesis is entailed by the verb in the text, but their arguments are unrelated, the result is UNKNOWN. An example of such a case is given below:

T: The cat ate a mouse.
H: The cat ate in the garden.

In all other cases of argument distribution in the case of Levin class similarity, the result is UNKNOWN. In case there is no match on the basis of the Levin classes, the algorithm attempts to extract entailment on the basis of the semantic description of verbs and then goes through the same steps. If none of the verbs match, either based on Levin classes or semantic descriptions, the algorithm returns UNKNOWN.

The algorithm described in this section has been implemented in part (the implemented part refers to argument entaiment), in the system used for the RTE-5 challenge; the 2010 version of the system, which was used for the RTE-6 evaluation campaign, implemented the predicate entailement section of the algorithm. The part that we have implemented concerns verb matching on the basis of VerbNet, as can be seen in Figure 2 given in chapter 4, which was used in conjunction with DIRT for coverage reasons. In the case of argument matching, the algorithm has gone through several versions, with the one given in this thesis being a more recent development. Because of this, even though an early version of argument matching has been included in the textual entailment system used in the RTE-5 challenge (see figure 1 in chapter 4, the extensions to the named entity recognizer and the WordNet module), the current version of the argument matching procedure, based on feature structure unification, is still in the process of being implemented for the next version of the textual entailment engine. Therefore, the results obtained in RTE-5 and the novelty detection task of RTE-6 prove the effectiveness of the algorithm.

3.3. Corpus analysis

As stated in the introduction, one of the key means of determining textual entailment, according to our intuition, is the analysis of the verbs. The goal of this section is to examine the usefulness of Levin classes, particularly in the form described in section 2, for the task of textual entailment, as a means of performing this comparison.

In order to determine the importance of using VerbNet for solving textual entailment, we have carried out a feasibility test on a corpus of 200 entailment pairs extracted from the RTE-5 test set, by manually examining each pair and determining the manner in which the correct solution can be deduced. The dataset used is typical for RTE data in terms of distribution of types of pairs; out of the 200 pairs taken into consideration, 100 (50%) are ENTAILMENT, 31 (15.5%) are CONTRADICTION, and 69 (35.5%) are UNKNOWN.

As a result of the analysis, every pair was classified into one of four categories:

- **VN** (**VerbNet**), which means that the application of the algorithm defined in section 3.2 leads to the correct result for the pair, on the basis of verbs in **T** and **H** sharing Levin classes, having similar semantic descriptions or verbs in **H** being semantic consequences of verbs in **T**, and having compatible argument structures;
- **NA** (**Not Attached**), which means that the algorithm defined in section 3.2 finds no verb in **T** that has the same argument structure as the verbs in **H**, or there are no verbs in **T** that are synonyms, consequences of or share a Levin class with the verbs in **H**;

- **NE** (**Not Exists**), which means that one of the key concepts in **H** is not found in **T**;
- **OS** (**Other Sources**), which means that the correct result can be deduced by other means than that of verbal comparison (e.g. coreference resolution or ontological knowledge). It is worth noting that, according to our analysis, most of the pairs that fall into this category are solved by extracting property sets for arguments and adjuncts and performing comparison over them.

Table 11 gives the distribution of the 200 pairs in these categories, in order of frequency:

Category	**Number**	**Percent**
VN	76	38%
OS	59	29.5%
NE	43	21.5%
NA	22	11%

Table 11: Distribution of solutions for feasibility study

As can be seen from the distribution of the solutions for the pairs taken into account for the feasibility study, the highest gain comes from the use of VerbNet, followed by other sources of entailment (almost all cases are instances of property set comparison). It also supports the intuition described in the introduction, as all the cases of NA can be reduced to mismatching predicates and all the cases of NE can be reduced to cases of mismatching property sets. For the rest of this section we will give a series of examples from the analyzed pairs that support the proposed methods.

3.3.1. Pairs solved using VN

This subsection contains a number of examples that support the use of VN as a tool in solving textual entailment. The most common use of VN is the alignment of verbs in the same Levin classes, as shown in the example below (pair 54 from the RTE-5 test set):

Example 1: Exact match over VN classes

```
T: MADAGASCAR'S constitutional court declared Andry Rajoelina as the
new president of the vast Indian Ocean island today, a day after his
arch rival was swept from office by the army. ...
H: Andry Rajoelina was proclaimed president of Madagascar.
```

The verb ***proclaim*** in the hypothesis is in class *say-37.7-1* in VerbNet, which also contains the verb ***declare*** in the text. These two verbs match in terms of arguments (***Andry Rojelina*** is the *Theme* as defined by VerbNet) and also in terms of adjuncts (***president of Madagascar***). It is also worth mentioning that this analysis also requires successful coreference resolution between "***Madagascar***" and "***vast Indian Ocean Island***". The fact that the two verbs matched over the text and the hypothesis are in the same Levin class, and that their argument structures match leads to ENTAILMENT, according to step 2.a. in the algorithm in section 3.2.

The use of a frame's syntactic description and a semantic decomposition is described in example 2 (pair 66 from the RTE-5 test set).

Example 2: Syntactic description and semantic decomposition

```
T: A court in Venezuela has jailed nine former police officers for
their role in the deaths of 19 people during demonstrations in 2002.
...
H: Nine police officers have had a role in the death of 19 people.
```

The verb ***have*** is part of the class *own-100*, which described syntactically as:

Theme1 V Theme2

and semantically as:

has_possession(E, Theme1, Theme2)

In this particular example, "***nine police officers***" is *Theme1* and *Theme2* is "***a role in the death of 19 people***"; therefore, the situation described in the text ("***their role***", which can be expanded, by means of coreference, to "***the policeman's role***") supports the hypothesis, and leads to ENTAILMENT, according to step 3.a. in the algorithm in section 3.2.

Because of the manner in which the VN classes are defined, it is possible that the verbs in the same class are not synonyms; furthermore, there are cases of antonyms in the same Levin class, as is the case in pair 92 given below.

Example 3: Antonymy in the same VN class

T: BEIJING — China has rejected Coca-Cola Co.'s $2.3 billion bid to buy a major Chinese juice producer, the Ministry of Commerce (MOC) announced Wednesday. The ruling is the first of its kind since China promulgated its anti-monopoly law last August. The proposed purchase was rejected on anti-monopoly grounds, MOC said on its website. "The bid may harm competition...in China's beverage market." The purchase of China Huiyuan Juice Group, the nation's largest juice maker, would have been the biggest foreign acquisition of a Chinese company to date.

H: Coca-Cola buys Huiyuan Juice Group.

In order to correctly solve this entailment pair the verbs ***reject*** and ***buy*** in the text need to be linked, in order to deduce the contradiction. The verb ***reject*** is part of the class *approve-77*; the case encountered in the text is syntactically described by the frame

Agent V Proposition

and semantically by

approve(during(E), Agent, Proposition)

Since the verb in question is opposed to the concept described in the semantic frame, the description becomes

not(approve (during(E), Agent, Proposition))

The second step of determining the correct solution is the analysis of the proposition, which is ***Coca-Cola Co.'s $2.3 billion bid to buy a major Chinese juice producer***; the class attached to the verb buy matches with the verb from the hypothesis. The combination of the two predicates gives the correct result for this pair, which is CONTRADICTION, according to step 3.b. in the algorithm in section 3.2. In order to correctly solve this pair, the anaphoric relation between “major Chinese juice producer” and “Huiyuan Juice Group” must also be determined.

In the cases where there is no link via Levin classes between verbs in H and T, the semantic description of the verbs can be used in order to determine synonymy, as seen in pair 126.

Example 4: Match by semantic decomposition

```
T: ..., and Fiat, the Italian car company that wants to acquire a
stake in Chrysler. ...
H: Fiat wants to gain possession of a stake in Chrysler.
```

Even though there is no verb in **T** that is in the same class (*get-13.5.1*) as ***gain***, the semantic description for this class,

has_possession(start(E), ?Source, Theme) transfer(during(E), Theme) has_possession(end(E), Agent, Theme) cause(Agent, E)

is identical to the semantic description for the class *obtain-13.5.2-1*, which contains the verb ***acquire*** from **T**. This match, coupled with a match over argument structures, leads to ENTAILMENT, according to step 3.a. in the algorithm in section 3.2.

VerbNet also allows for the establishing of relations between verbs on the basis of the semantic description of each frame. This can be seen in the relation between the verbs assassinate and die, as can be seen in pair 41.

Example 5: Establishing consequence relations

```
T: Varun and his estranged cousins are the grandchildren of Indira
Gandhi, the former prime minister who was assassinated in 1984. Her
two sons - Rajiv Gandhi, who became premier after Indira's
assassination and Sanjay Gandhi who died in an aircraft crash while
doing aerobatics - married women with very different personalities.
Varun's mother, Maneka, widow of Sanjay Gandhi, is headstrong,
voluble, and strident. Rahul and Priyanka's mother is the Italian-born
Roman Catholic, Sonia Gandhi, who was widowed by Rajiv's
assassination, was more demure as a daughter-in-law.
H: Indira Gandhi died doing aerobatics.
```

The semantic description for the VN class *murder-42.1*, to which ***assassinate*** belongs to, is *cause(Agent, E) alive(start(E), Patient) not(alive(result(E), Patient))*

and that for the class *disappearance-48.2*, the class of ***die***, is

disappear(during(E), Theme)

Since ***die*** (***not alive***) is a troponym of ***disappear*** (according to WordNet), and the arguments of the two semantic descriptions match (*Theme* and *Patient* are similar), it can be deduced that the frame attached to the verb ***die*** is a consequence of the frame attached to the verb ***assassinate***. The end result of the entailment pair is CONTRADICTION, according to step 3.b. in the algorithm in section 3.2, as the predicate in the hypothesis has an adjunct (***doing aerobatics***) that is in contradiction with the adjuncts of the predicate assassinate.

3.3.2. Pairs solved using NE and NA

In the case of the three way task for Recognizing Textual Entailment, one of the difficulties is the recognition of UNKNOWN cases. According to the intuition on the basis of which the analysis has been carried out, UNKNOWN cases are due to two main reasons: one or more of the entities in **H** are not present in **T** (**NE**), or all the information in **H** is present in **T**, but is not linked by any one predicate (**NA**).

An example of entities present in the hypothesis which are not semantically and syntactically linked in the same way in the text can be seen in example 6 (pair 82).

Example 6: Entities from H unrelated in T

T: Currently, there is no specific treatment available against dengue fever, which is the most widespread tropical disease after malaria. Sanofi Pasteur is collaborating with the Communicable Disease Center in Singapore and the Pasteur Institute in Vietnam to conduct these clinical studies in children and adults. "Controlling the mosquitoes that transmit dengue is necessary but not sufficient to fight against the disease. A safe and effective vaccine has been long awaited to prevent dengue epidemics," said Professor Leo Yee Sin, director of the Communicable Disease Center in Singapore. "Clinical studies in Singapore are critical steps to advance the development of a vaccine

```
for the prevention of dengue in Asia. We are happy to contribute to
scientific re-search that would benefit the entire region."
```

H: Malaria is the most wide-*spread* disease transmitted by mosquitoes.

As can be seen in the text, ***malaria*** appears just once, and not in any way linked to the notion of ***mosquitoes*** in the text (even though WN defines malaria as a disease transmitted by mosquitoes, this is not specifically mentioned in the text), although it is also mentioned in T; this is thus an UNKNOWN case, according to step 2.c. of the algorithm in section 3.2.

Another way of extracting UNKNOWN cases is to have an entity in the hypothesis that cannot be found in the text (**NE**), as can be seen in example 7 (pair 21).

Example 7: An entity from H not found in T

T: I was nearly charged with petty theft for pilfering coffee at the illustrious Hippodrome Building. But lest I be judged too quickly, I must convey the sublimity of the fourth floor's coffee machine. Harry Houdini performed at the Hippodrome, at 1120 Avenue of the Americas near 44th Street. Many of the best and most famous performers of the time appeared there. It was one of the biggest and most successful theaters of its time, capable of accommodating 5,200 people.

H: Harry Houdini was a magician.

This pair is an UNKNOWN case because there is no mention of the entity “magician” in any form.

After the analysis, all of the 22 NA and 43 NE cases were found to point to UNKNOWN cases, and account for 65 of the 69 total UKNOWN cases. The remaining four cases are due to the fact that the predicates used in the hypothesis are not found in the text, as can be seen in example 8 (pair 105).

Example 8: Determining UNKNOWN using verb matching

T: The company also faced shortages of skilled riveters, the archives showed. Dr. McCarty said that for a half year, from late 1911 to April 1912, when the Titanic set sail, the company's board discussed the problem at every meeting. For instance, on Oct. 28, 1911, Lord William Pirrie, the company's chairman, expressed concern over the lack of

```
riveters and called for new hiring efforts. In their research, the
scientists, who are metallurgists, found that good riveting took great
skill. The iron had to be heated to a precise cherry red color and
beaten by the right combination of hammer blows. Mediocre work could
hide problems.
```

H: `Titanic sank in 1912.`

This is not a typical **NA** case because of the predicate ***set sail*** in the text, which has as arguments the exact entities that can be found in the hypothesis. Therefore, the correct solution of this pair is dependant exclusively on verbs. The VerbNet class of the verb ***sink*** is *other_cos-45.4*, and has the semantic description

state(result(E), Endstate, Patient)

The only other verb in T that is related in any way to this one is ***heat***, which is actually in the same class as ***sink***, but has no arguments that are linked in any way to ***Titanic*** or ***1912***, thus leading to UNKNOWN, according to step 3.c. of the algorithm described in section 3.2. This is an interesting case, also because the class *other_cos-45.4* is very large and contains verbs that have significantly different senses. Despite all this, there is no real danger of incorrectly matching two (semantically) unrelated predicates from this class, as the widely different senses also require significantly different arguments (this observation applies to most of the VerbNet classes in this situation).

3.3.3. Pairs solved using OS

As stated above, **OS** (Other entailment Sources) refers to the solving of textual entailment by other means than verb matching (mostly, the analysis of entities in order to attach the feature sets so that entailment analysis can be carried out). This is in accordance to the intuition given in the introduction, according to which entailment can be determined through feature set unification. The difficulty of this approach is the classification of attributes; this is solved by the use of an ontology (preliminary tests showed that both WordNet (Fellbaum 1998) and SUMO (Niles and Pease, 2001) are useful for this). An example of using world knowledge in order to determine entailment is given in the analysis of example 9 (pair 7).

Example 9: Solving entailment with word knowledge (part_of)

```
T: Global technology giant IBM is in talks to buy Sun Microsystems in
a deal that would expand its server market share, the Wall Street
Journal reported Wednesday. IBM may pay as much as US$6.5 billion in
cash for Sun, the newspaper reported on its Web site, without naming
its sources. That amount of money would be nearly double Sun's closing
share price on Tuesday of $4.97 per share. The report cautioned that
while the two companies are holding discussions, a transaction may not
occur.
H: The price of Sun Microsystems is $4.97.
```

World knowledge is used in this case in order to determine that ***share*** is a part of a ***company***, as determined from SUMO. According to this knowledge, it follows that the price of a ***company*** is greater than the price of a ***share***, which leads to CONTRADICTION.

Another example of using world knowledge in order to solve entailment can be found in example 10 (pair 191).

Example 10: Solving entailment with word knowledge (is_a)

```
T: Ernie Barnes, whose drawings and paintings of athletes, dancers and
other figures in motion reflected his first career as a professional
football player, died on Monday in Los Angeles. He was 70. The cause
was complications of a blood disorder, his personal assistant, Luz
Rodriguez, said. Mr. Barnes was an offensive lineman in the old
American Football League, playing four seasons in the 1960s for the
New York Titans, the San Diego Chargers and the Denver Broncos. He
would often say later that even during his playing days, his heart was
more in the painting and sketching he had been doing since he was a
child.
H: Ernie Barnes was an athlete.
```

In this case, WordNet contains the information that ***offensive lineman*** is a type of ***athlete***, which leads to the classification of this pair as ENTAILMENT.

Also, copulative verbs need to be seen as a form of assignment of properties; the hypotheses of both examples in this subsection have a copulative verb, which is not analyzed by means of VerbNet but which is treated as a link between an entity and an attached property.

A very important issue in terms of feature set extraction for entities is the solving of coreference chains. This can be helped by VerbNet because the chance of two entities corefering is increased in the case of entities that have the same argument position with regards to verbs belonging to the same Levin class, as can be seen in the example below.

Example 11: Solving entailment by means of coreference resolution

```
T: The BBC has seen documents alleging that a former rebel leader
indicted for war crimes is playing a leading role in a mission
involving the UN in DR Congo. The documents appear to prove that Gen
Bosco Ntaganda is taking an active part in the mission's chain of
command, a BBC correspondent in the country says. The UN-Congolese
force is fighting Hutu rebels in the eastern DR Congo. The force says
Congolese authorities have given assurances that Gen Ntaganda was not
involved in joint operations.
H: Bosco Ntaganda was a rebel leader.
```

The corefering entities are ***rebel leader*** and ***Bosco Ntaganda*** and they are both *Agents* of the verbs *play* and *take part* respectively, which are part of class *performance-26.7*, thus leading to ENTAILMENT.

The use of feature set unification for determining entailment can be seen in example 12 given below:

Example 12: Solving entailment by means of feature set unification

```
T: Justice Ian Josephson was still to pass judgment on Ajaib Singh
Bagri, 55, who along with Malik had faced charges of first-degree
murder, conspiracy to commit murder and attempted murder in the
downing of Flight 182.
H: Ripudaman Singh Malik had faced charges of murder and conspiracy.
```

On the basis of the algorithm described in section 3.2, step 2.a. determines that the verb in the hypothesis has the same class as a verb in the text (in this case, the verbs are identical "*had faced*"). On the basis of existing verb matching, the algorithm then attempts to perform argument matching in order to determine entailment. The verb in the hypothesis has 2 arguments; the subject is "*Ripudaman Singh Malik*", which matches with the subject in the text, "*Malik*" (previous information available in the source document proves that there exists a coreference reation between the two entities). The direct object in both cases is a compound structure, but the unification of these structures is successful, and is equal to the feature structure given in the text. Therefore, the result for this entailment pair is ENTAILMENT.

3.4. Conclusions

In this section we have described a novel way of solving textual entailment using predicational semantics and feature structures. The analysis has been carried out on a corpus of 200 entailment pairs extracted from the RTE-5 test set.

For determining predicational semantics we have explored the utility of VerbNet, which was able to solve 38% (76 pairs) of the set taken for analysis. Another 29.5% of the test set (59 pairs) can be correctly solved by using property sets defined on the basis of coreference, copulative verbs, word knowledge (ontologies) and various grammatical constructions (such as appositions). We have also discovered that, on the given corpus, UNKNOWN cases behave very regularly, as most of them (65 out of 69) are solved as either **NA** cases (the information in the hypothesis is not linked by any predicate in the text) or **NE** cases (there is one entity in the hypothesis that does not exist in the text).

The feasibility study given in this section also provides the motivation for the use of VerbNet for textual entailment. First of all, the use of VN for textual entailment is novel, as there are, to our knowledge, no attempts at using predicational semantics in general, and VN in particular, to create a deep semantic representation of entailment pairs. The study also proves the usefulness of such an approach, as it has shown that 38% of entailment pairs can be solved using the information in VerbNet, and 29.5% can be solved by means of argument unification. We have also given an algorithm that solves textual entailment on the basis on verb knowledge provided by VerbNet and ontological knowledge provided by taxonomies such as WordNet.

The problem of using VN for textual entailment is a complex one, however, as can be seen in the example below (pair 145):

Example 13: Entailment based on inference chains

T: If 20 multiple sclerosis patients were in a room, all would exhibit different symptoms, she says. "The disease is not clear and not set. It's highly unpredictable." In particular, in the relapsing-remitting

```
stage, there are symptoms such as fatigue, blurry vision, numbness or
tingling that the patient feels but are not noticeable to others. As a
result, outsiders may not be aware of the extent of the patient's
suffering. "Sometimes, they think if you're not in a wheelchair, M.S.
must not be that bad. That's not true," Caesar observed. In Florida,
20,000 individuals have been diagnosed with the disease. Nationwide,
that number is 400,000. Although it is still in the basic research
stage, stem cell therapy shows promise, says Caesar. The research
focuses on getting stem cells to regenerate the nerve-insulating
myelin.
```

H: `Stem cells could be useful to treat multiple sclerosis.`

The solution for this pair (which is ENTAILMENT) can only be extracted after performing a series of complex inferences on the basis of the text; "***stem cell therapy shows promise***" is an incomplete proposition, as it does not have a *Recipient*, which is, in this case, "***multiple sclerosis***". The semantic description attached to "***show***" is

indicate(during(E), Cause, ?Recipient, Topic)

and links the *Recipient*, not realized in this frame but linked to "***multiple sclerosis***", to the *Cause*, which is "***stem cell therapy***" and the Topic, "***promise***". This linking of the three concepts is similar to that in **H**, as "***shows promise***" is similar to "***useful***", and therefore leads to ENTAILMENT. However, this extraction of the *Recipient* is difficult, and hinges on the identification of the fact that it is not realized in the surface form of **T**.

This method is also feasible in the sense that the required analyses can be done with the aid of current generation parsing techniques, such as functional dependency parsing and semantic role labelling, which are accurate for English. Also, the semantic descriptions attached to verb classes in VerbNet are, in fact, atomic propositions grouped into equivalence classes, which greatly reduces the amount of work required to match up predicates.

The effectiveness of the method we have proposed for solving textual entailment has been proven in two ways: firstly, the results obtained in the last two RTE challenges (particularly RTE-5) have shown that the systems enhanced by means of this method are among the top scoring systems in the world. Secondly, the feasibility study given in this chapter has shown that our method has the capability of improving a state of the art system by correctly solving entailment cases that were missed without decreasing performance (this analysis was carried out over the results obtained by the top scoring system in RTE-5; existing correct solutions were validated by our algorithm, and incorrect solutions, such as those obtained for pairs 1, 2, 102,105, to give just a few).

As future work, we intend to further expand a series of resources linked to VerbNet, particularly to attach semantic descriptions to the selectional restrictions and predicates, in order to then automatically extract relations between frames and verb classes.

A major advantage of using VN is that it is aligned with a series of resources such as FrameNet and WordNet, which means that creating a similar resource for Romanian can be done quickly (the main difficulties arise from the alignment of the syntactic frames in English and Romanian). Given a Romanian VerbNet, the algorithm put forward in this section could be used to solve entailment pairs in Romanian without the need for modifications. The same can be said for the world knowledge bases (particularly SUMO), as they are also aligned to WordNet, making it much easier to port such resources for Romanian.

Because of the advantages described above, our approach can solve textual entailment in a language independent context, given the appropriate resources: WordNet for the given language, a version of VerbNet (which can be created on the basis of WN), and parsing tools for that language (POS tagger, syntactic parser, and, possibly, a semantic role labeller). Chapter 5 describes two approaches to extending the Romanian version of WordNet; given a WN version that has significant coverage, the algorithm described at the beginning of this chapter can be implemented for solving textual entailment for the Romanian language. The proposed extension methods are based on the electronic format of the Romanian Language Thesaurus Dictionary (eDTLR), obtained within the eDTLR grant, PNCDI II Project No. 91_013/18.09.2007, to which we had significant contributions. Our contribution towards obtaining the electronic format of the Romanian Language Thesaurus Dictionary, together with the methods developed and implemented for obtaining the eDTLR are described in detail in chapter 5.

3.5. Comparison with Current Entailment Methods

The inherent difficulty of recognizing textual entailment arises from the fact that it attempts to address the issue of natural language variability; informally, natural language variability is means conveying the same semantic information with possibly significantly different surface realizations of the text. It is our sincere opinion that this can only be properly addressed by a deep semantic understanding of natural language, and that this deep semantic understanding of natural language can be reached by turning to the basic building blocks of human utterances, predications and their arguments. We have based the assertions made in this chapter on the results obtained by the systems described in section 2 in the RTE competitions, which give an experimental setting for the methods discussed, because the unified evaluation framework availableallows for the comparison of significantly different approaches.

Given the reasons stated above, it can be said that textual entailment solving methods based on lexical matching (section 2.1), syntactic graph distance (section 2.2) and tree edit distance (section 2.3) cannot provide a sufficiently detailed deep semantic representation of the text and hypothesis, and it is our opinion that, because of this, our method shows more room for improvement and greater flexibility in adapting to new entailment settings. Also, pure logical representations of natural language utterances are difficult to obtain, thus hindering the ability of such systems as those described in section 2.4 to obtain deep semantic representations of language.

It can be argued that some of the more successful approaches for solving textual entailment consists of using atomic propositions (described in section 2.5), as they address, at least in part, the notion of predicates in natural language. The difficulty of this approach, however, is that it treats the text and the hypothesis as bags of atomic propositions, which does not accurately represent the deep semantic meaning of natural language, as it disconnects arguments of the same predicate from one another.

Because of the same reasons described above, machine learning based systems (section 2.6) also have difficulties in creating deep semantic representations of natural language; since machine learning leads to emulating behaviour described in the training data, it can be said that such a system creates accurate feature representations of entailment pairs, and then draws a conclusion upon them without actually understanding the utterances, but based on previously annotated data. This difficulty translates into a lack of flexibility when encountering previously unseen linguistic phenomena.

It is our belief that approaches based on discourse knowledge and analysis (section 2.9), coupled with entailment rules (section 2.7), either manually created or automatically extracted from semantic resources such as FrameNet are among the best methods for solving current entailment problems, but that it also needs to be coupled with predicate structures in order to create a robust entailment engine. Because of these facts, we believe that the approach described in this thesis is a viable one for solving the problem of textual entailment in the general case, as it is both scalable and robust.

The approach we have proposed in chapter 3 is a first step towards creating a deep semantic understanding of natural language. Therefore, we are convinced that the method put forth in this thesis advances the state of the art in textual entailment towards a semantic based interpretation of language, an idea which is closer to the initial definition of textual entailment, given in (Dagan et al., 2005): “We say that a text T entails a hypothesis H if, typically, a human reading T would infer that H is most likely true”.

4. System Description, Experimental Results and Analysis

The system used for solving textual entailment is based on the one described in (Iftene, 2009), which is an improved version of the system initially described in (Iftene and Balahur-Dobrescu, 2007) and (Iftene, 2008). On the conceptual level, the system uses rules to map every entity from the hypothesis to at least one hypothesis from the text by using extensive lexical-semantic knowledge sources such as WordNet, VerbOcean, Wikipedia derived lexical information, etc. The mapping process assigns a fitness score to each word in the hypothesis on the basis of the rules used; these local fitness values are used to calculate a global fitness value. Local fitness values are defined on the interval [0, 1], on the basis of word correlation (1 means maximum correlation and 0 means that the words are not correlated at all). Depending on modifiers applied to the words (negation of verbs, for example), this fitness value may change. Global fitness is computed as a function of the local fitness of each hypothesis element; in the end, a fitness value is assigned to the entailment pair that ranges from -1 to 1. The closer the global fitness value is to the ends of the interval, the more the hypothesis and the text are correlated; positive values are characteristic of agreement, while negative values are characteristic to disagreement. In order to solve the three-way textual entailment, the system relies on two empirically computed thresholds that separate the high correlation pairs from the low correlation pairs (two-way textual entailment can be derived from three-way textual entailment). Section 6.1 below describes the general system architecture and development.

4.1. General System Architecture

In this section we will give a brief description of the basic version of the system used for the RTE-5 and RTE-6 evaluation campaign. This system was first described in (Iftene, 2009), and then further improved in (Iftene and Moruz, 2010), and was used as a starting point for applying the improvements and algorithms described in chapter 3 of this thesis.

4.1.1. Pre-Processing

The preprocessing module is concerned with the transformation of the raw text received as input into a structured format, upon which entailment rules can then be applied.

Since the data sets contain text that may have grammatical or spelling errors, and also because some of the processing tools behave poorly when encountering certain phenomena, such as contractions, some modifications are performed over the input text (Iftene and Moruz, 2010). Thus, all contractions are expanded: "*hasn't*" replaced with "*has not*", "*isn't*" replaced with "*is not*", "*couldn't*" replaced with "*could not*", etc. The meaning remains the same after this transformation, but the quality of the MINIPAR (Lin, 1998) output is considerably improved. Also, before sending the text to LingPipe[14], some punctuation signs like quotation marks "", brackets (), [], {}, commas, etc. are padded with spaces (replaced with the initial character surrounded with space characters). Again, the meaning of the text is the same, but the quality of the LingPipe output is improved after this transformation.

After the preparation step, the text and the hypothesis are parsed with MINIPAR. For those cases in which MINIPAR does not identify any verb in the processed sentence, the system employs TreeTagger for identifying, with a higher degree of precision, the Part-Of-Speech (POS) and replaces the incorrect POS determined by MINIPAR. This step is very important, especially in the case of verbs, because our algorithm begins its mapping with verbs, and builds the comparison on the basis of verb arguments.

In parallel, the result obtained after preparation is processed by LingPipe, in order to identify named entities (NEs). In the case of the Named Entities of type JOB and LANGUAGE, GATE (Cunningham et al., 2001) is used as an additional resource, because its gazetteer contains finer-grained classes of entities, which considerably increase the accuracy of NE extraction.

4.1.2. Main Module

The purpose of the main module is to map all the nodes from the hypothesis syntactic tree to at least one node in the syntactic tree of the text, in a similar manner as that described in (Iftene, 2008), (Iftene and Moruz, 2010), (Iftene and Moruz, 2011). The mapping between entities can be done either *directly* (when entities from hypothesis tree exist in the text tree, in the sense that we can identify a lexical match between entities in the hypothesis and the text) or *indirectly* (when entities cannot be mapped directly and require transformations using external resources). The mapping of each node yields a fitness value, on the basis of the transformations involved in order to carry it out (exact match yields a fitness of 1, for example, and antonyms yield a fitness of -1), which indicates the similarity between entities of the text and the

[14] LingPipe: http://www.alias-i.com/lingpipe/

hypothesis. Using the local fitness values, an extended local fitness is computed, and then, using all partial values, the system calculates a normalized value that represents the global fitness. When an entity from the hypothesis can be mapped to more entities from the text, the system selects the mapping which maximizes global fitness.

The global fitness value is then used to determine the relation between text–hypothesis pairs. This is done by comparing the global fitness of the entailment pair to two thresholds computed on the basis of the training corpus; these thresholds are between -1 and 1, and partition the solution interval into three separate intervals: entailment cases are characterized by global fitness in the top interval, contradiction cases are characterized by global fitness values in the bottom interval, and unknown cases have global fitness values that fall in the middle interval (Iftene, 2009).

4.1.3. Mapping Rules for Entailment Cases

In order to determine the global fitness for a given entailment pair, the entailment engine attempts a mapping of the nodes from the syntactic tree of the hypothesis to the nodes of the syntactic tree of the text (Iftene, 2008), (Iftene and Moruz, 2010), (Iftene and Moruz, 2011). For every node from the hypothesis tree which can be mapped directly to a node from the text tree, the local fitness value is 1 (which represents the maximum value). When direct mapping is not possible (no direct lexical match), the system applies transformations to the unmatched hypothesis node, based on external knowledge sources, so that it becomes more similar to some node in the text. For verbs we use DIRT (Lin and Pantel, 2001) and transform the hypothesis tree into an equivalent one, where the verb node is replaced with an equivalent form. This is supplemented by the use of VerbNet[15] in order to determine the relation between the verbs in the hypothesis and the text, as described in (Moruz, 2010). This is the case of the example below, where in the text we have "*An English-born blues legend, passed away…*" and in the hypothesis we have "*A musician has died…*". After using this resource, the hypothesis changes into "*A musician passed away*" and in this form it is easier to compare the text and hypothesis and, in the end, the value of the global fitness score is increased.

[15] VerbNet: http://verbs.colorado.edu/~mpalmer/projects/verbnet.html

In the case of named entities, the system either uses an acronym database or obtains information related to it from background knowledge (Iftene and Balahur-Dobrescu, 2008). Apart from the relations between acronyms and full names relations such as *part-of* (relation between Texas and USA) are also taken into account.

For nouns and adjectives WordNet (Fellbaum, 1998) is used, along with some of the relations from eXtended WordNet[16] to look up synonyms, which are then mapped to nodes from the text tree. Also, on the basis of the WornNet taxonomy, we perform argument matching, as described in chapter 3.

For every transformation with DIRT or WordNet, it is considered that the value of the local fitness is the value of the term similarity in the respective resources. If the mapping was carried out using background knowledge or the acronym database, the value of the local fitness is considered to be 1.

In the case of numerical data the mapping process is not as straightforward as for nouns, for example, and some special situations need to be taken into account (Iftene, 2009). There are cases in which, even if the numbers from the text and the hypothesis do not correspond, certain quantifiers may change their meaning enough for a positive match. For solving these cases, the system creates intervals for both the text and the hypothesis expressions; if the interval from the text is contained in the interval from the hypothesis, it awards a local fitness value of 1. The quantifiers are taken from a list which contains expressions such as "*more than*", "*less than*", or words such as "*over*", "*under*", etc, which describe intervals that are unbound on one end (Iftene, 2008), (Iftene and Moruz, 2010). In the case of intervals that are bounded at both ends, a local fitness with a value of 1 is awarded only if the interval from the hypothesis is included in the interval from the text (e.g. T: "John trains from Monday to Friday", H: "John trains from Tuesday to Friday"), in opposition to the previous case, where the intervals in the text and the hypothesis were unbound at one end.

If any of the numbers in the text or the hypothesis has an attached unit of measure, it is always kept, as it is possible to find the same numbers in the text and the hypothesis, but to have those numbers referring to different entities:

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

```
T: At least 14 people have been killed in a suicide bomb attack; government officials were among the 35 injured.

H: 35 government officials were injured.
```

4.1.4. Mapping Rules for Negative Entailment Cases

First, if after all checks are made we cannot map a node from the hypothesis syntactic tree, the value of the node's local fitness is weighed down; the weight depends on the type of entity in question, as some entities are more important to determining entailment (for example, since named entities are very indicative of entailment, the failure to map a NE from the hypothesis to the text inserts a very large penalty, while the failure to map a common noun results in a lighter penalty). Also, because stop words from the hypothesis ("*the*", "*an*", "*a*", "*at*", "*to*", "*of*", "*in*", "*on*", "*by*", etc.) artificially increase the value of global fitness, they are not taken into consideration for the purpose of calculating the value of the global fitness (Iftene, 2008), (Iftene and Moruz, 2010), (Iftene and Moruz, 2011).

Since *No Entailment* cases can be further separated into contradiction and unknown cases, the system employs different sets of rules to identify these distinctive cases and assign the corresponding values for local fitness (in general, rules designed for finding contradiction tend to weigh down local fitness values with negative numbers, while rules designed for finding unknown cases use weights that are closer to 0)

The most indicative element for contradiction is negation for verbs. To every verb from the hypothesis the system attaches a Boolean value which indicates whether that verb is negated or not. For determining negation, the verb's subtree is examined for words such as "*not*", "*never*", "*cannot*", etc. For each of these words it successively negates the initial truth value of the variable attached to the verb, which is initialized with "*false*".

For determining contradiction, several situations are considered, the most common of which is the negation of a semantic equivalent of a verb from the text with a verb from the hypothesis. In the example below, the text is "*Movie studio company, New Line Cinema has announced that movie director Peter Jackson will never be allowed to work on another New Line films*." and in the hypothesis "*New Line wants to work with Peter Jackson*.". Another type of contradiction case is that of long infinitive verbs preceded by words such as "*refuse*", "*deny*", "*ignore*", "*plan*", "*intend*", "*proposal*", "*able*", etc.

Contradiction is also determined on the basis of the *semantic opposition* relation between synsets that contain words from the text and synsets that contain words from the hypothesis. For determining semantic opposition, the system uses the [*opposite-of*] relation from VerbOcean (Chklovski and Pantel, 2004), the antonymy relation from WordNet and the inter-verb relations extracted using VerbNet (all these resources are used for greater coverage). The domain of the antonymy relation is broadened by combining synsets and antonyms in WordNet or opposites from VerbOcean. For words from the hypothesis which cannot be mapped to words from the text using either synonymy or antonymy, the system considers the set of antonyms for their synonyms and then checks if any entity from this new set can be mapped to an entity in the text.

In some situations, the co-occurrence relation from DIRT is an antonymy relation (the scores in DIRT are in fact more similar to co-occurrence scores), and for this reason an extra check of DIRT relations is performed in order to see if there exists an antonymy relation between them in either WordNet or VerbOcean. For all identified contradiction cases, because the system applies the greatest penalty possible, the final answer for the considered pairs will be "*Contradiction*"; however, contradiction cases need to be clear, in the sense that a verb in the hypothesis is a negation of a synonym of a verb in the text, or one of the entities in the hypothesis is an antonym of an entity in a similar position in the text, so that the system solves contradiction consistently (Iftene, 2008), (Iftene and Moruz, 2010), (Iftene and Moruz, 2011).

If the text or hypothesis contains modifiers such as "*may*", "*can*", "*should*", "*could*", "*must*", "*might*", "*infrequent*", "*rather*", "*probably*", etc., ahead of already aligned verbs (verbs are aligned using the rules described for the positive cases: synonymy, co-occurrence, consequence, etc.), the penalties are not decisive in establishing the final answer, which is obtained only after computing global fitness; this is because these types of words weigh down the local fitness of the modified word by a smaller amount, and usually point to an unknown case.

In the case of named entities, however, the solution chosen is different, and relies on the previously stated observation that all the named entities in the hypothesis need to be aligned with entities in the text. If even after using the acronym database and the external knowledge base there exists a named entity in the hypothesis that cannot be mapped to an entity from the hypothesis to an entity in the text, the conclusion is that the available data is not sufficient for determining a precise answer for the given entailment pair, and the result for the pair is "*Unknown*".

An exception to the named entity rule presented above is the case when the entity is a first name, in which case we only insert a penalty in the global fitness; indeed, it can be argued that, for this pair, the alignment holds because the named entities in the text and the hypothesis are in a coreference relation.

T: `A man is accused of killing Ms. Zapata.`

H: `Angie Zapata has been killed.`

In the general case, if there are enough entities in the hypothesis that are not aligned with elements in the text, or if enough of the aligned entities are preceded by modal modifiers, the global fitness for the pairs tends toward 0 (which means unknown), a fact which confirms the experimental observation that the text and the hypothesis in unknown pairs are relatively unrelated on the lexical-semantic level.

For the Main Task in the RTE-6 challenge it was no longer necessary to separate the no entailment cases into "*Unknown*" and "*Contradiction*". Because of this, the threshold separating these two sets of pairs is ignored, and the same answer is given in both cases, but it is important to note that the option to separate unknown and contradiction is still available.

The RTE-5[17] track at TAC 2009 continues the previous RTE Challenges that have aimed to focus research and evaluation on the underlying semantic inference task. The Fifth Recognizing Textual Entailment Challenge (RTE-5) brought two significant changes compared to RTE-4: 1) the average length of the Texts was greater, and 2) texts came from a variety of sources and without any additional corrections or simplifications as compared to the source documents. (Iftene and Moruz, 2010) describes the system in detail.

The system used for RTE-5 represents an improvement over the previous systems from RTE3 (Iftene and Balahur-Dobrescu, 2007) and from RTE-4 (Iftene, 2008). Also, we added new modules and used new semantic resources to deal with the new changes, and also with the aim of better identifying Unknown cases. Figure 1 shows the system we have employed, with the newly added components highlighted in grey (Iftene and Moruz, 2010):

---

[17] http://www.nist.gov/tac/tracks/2009/rte/

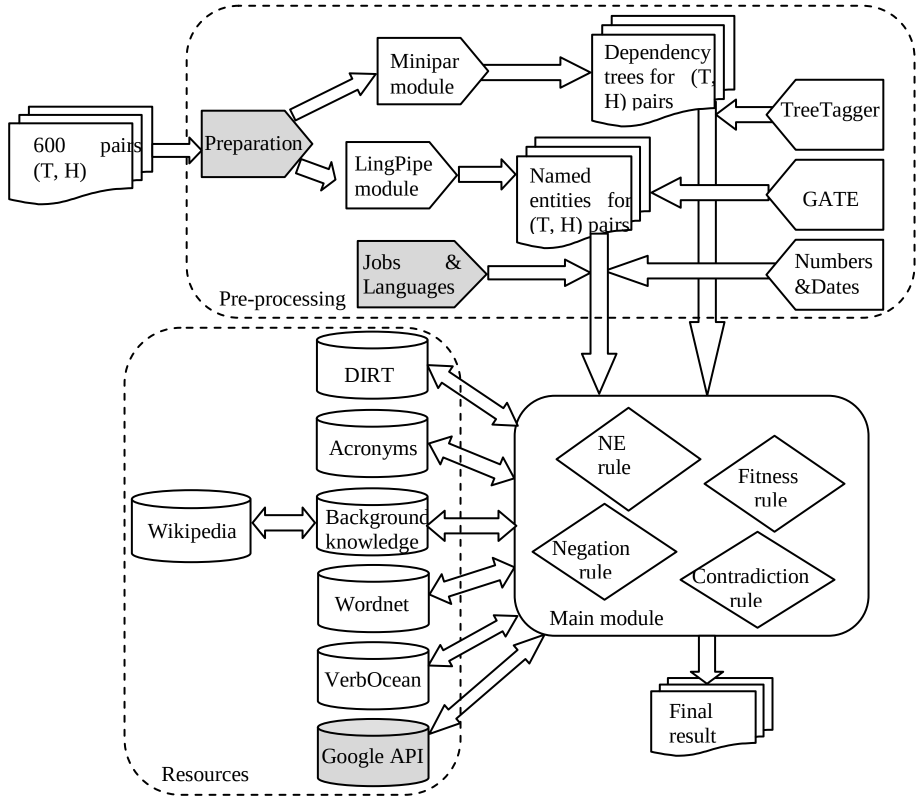

Figure 1: RTE5 System architecture

The 6th Recognizing Textual Entailment Challenged changed significantly compared to previous challenges, by moving the task to a more realistic scenario. According to the RTE-6[18] guidelines, by building on the promising outcome of the RTE-5[19] Pilot Search Task, RTE-6 has two goals:

- *to advance the state of the art in RTE*, by proposing a data set which reflects the natural distribution of entailment in a corpus and presents all the problems that can arise while detecting textual entailment in a natural setting, such as the interpretation of sentences in their discourse context;
- *to further explore the contribution that RTE engines can make to Summarization applications*. In a general summarization setting, correctly extracting all the sentences

[18] RTE-6: http://www.nist.gov/tac/2010/RTE/

[19] RTE-5: http://www.nist.gov/tac/2009/RTE/

entailing a given candidate statement for the summary (similar to Hypotheses in RTE) corresponds to identifying all its mentions in the text, which is useful to assess the importance of that candidate statement for the summary and, at the same time, to detect those sentences which contain redundant information and should probably not be included in the summary. Furthermore, if automatic summarization is performed in the Update scenario (where systems are required to write a short summary of a set of newswire articles, under the assumption that the user has already read a given set of earlier articles) it is important to distinguish between novel and non-novel information. In such a setting, RTE engines which are able to detect the novelty of H's can help Summarization systems filter out non-novel sentences from their summaries.

The basis for the system used in RTE-6 was the system created for RTE-5 (Iftene and Moruz, 2010), which was further refined and honed. Most of the modifications consist of tweaking various thresholds and extra pre-processing of the pairs and of a new data source that has also been added, in the form of VerbNet, as well as modifications to rules in order to accommodate the new knowledge source. The system consists of a preprocessing module and a decision module, which contains the rules that determine the value of the entailment pair. A schema of the system is given in Figure 2 below. Broadly, the system first runs the input through a preprocessing stage, which expands contractions, recognizes NEs, performs dependency parsing and recognizes dates. The processed input is then fed into the decision module, which, using resources such as WordNet, VerbOcean, VerbNet, and so on, makes a decision upon the entailment value of the given pair. The dotted resource module in figure 2 shows where the current method for argument matching is being added, in preparation for the RTE-7 challenge. Previous versions of argument matching procedures were included in the WordNet module in the form of entailment rules.

In sections 5.2 and 5.3 below, we give a detailed description of the performance of our system. The results described are obtained on the data sets of the latest Recognizing Textual Entailment Challenges; the reasons for choosing these date sets are twofold: the RTE challenge provides a framework for testing textual entailment engines, allowing for the comparison of different systems, and the data sets provided are intended to advance the current state of textual entailment, by incrementally proposing more difficult problems.

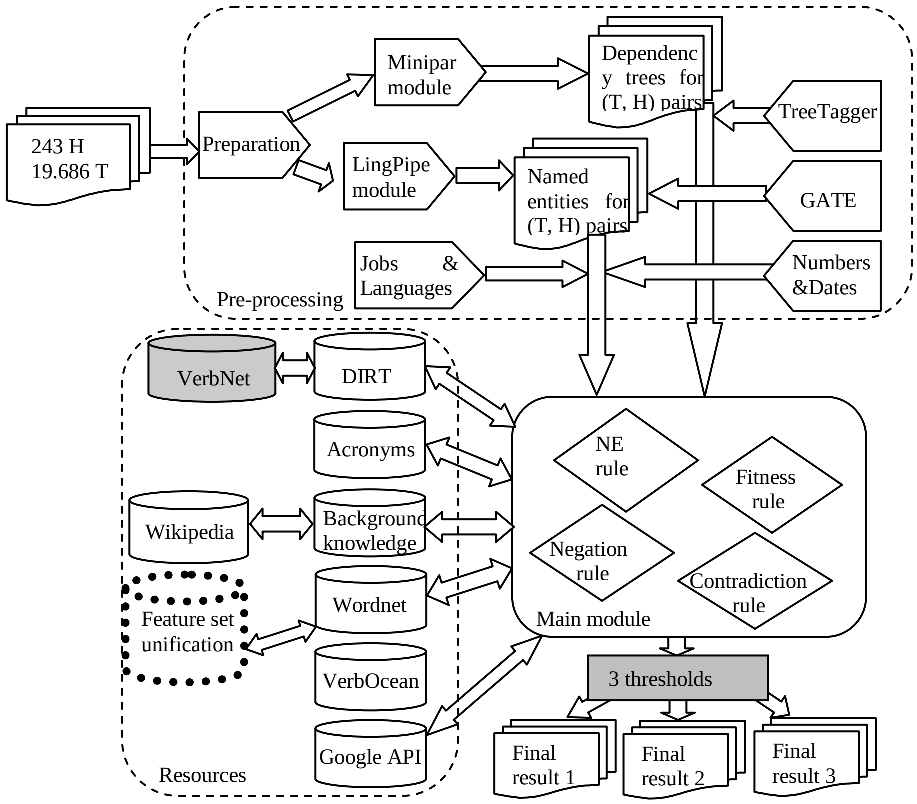

Figure 2: RTE-6 System architecture

4.2. Results in RTE-5

This section describes the results obtained in the 5th RTE challenge. (Iftene and Moruz, 2010) describe the participation at large; it is worth noting that our entry for RTE-5 won both main tasks (the two and three way challenges) by a margin of 5%, and that our solution for the pilot task was fourth overall (Bentivogli et al., 2010). The distributions of our answers in the 3-way task are presented below:

Answer Type	In Gold	Correct offered by our system	Total offered by our system	Precision	Recall	F-measure
Entailment	300	260	379	68.60%	86.67%	76.58%
Contradiction	90	22	44	50.00%	24.44%	32.84%
Unknown	210	128	177	72.32%	60.95%	66.15%
Total	600	410	600	68.33%		

Table 12: Results in RTE5 on 3-way task

As we can see, the highest precision is for the *Unknown* case, 72.32 %, and the lowest precision is for the *Contradiction* case, 50.00 %. Also, we can see that the highest recall and F-measure are obtained for the *Entailment* case 86.67 % and 76.58 % and the lowest are for the *Contradiction* case, 24.44 % and 32.84 %. The system offers many *Entailment answers* and catches almost all possible *Entailment* cases; also the system offers a lower number of *Contradiction* and *Unknown* answers, but most of these are correct.

For the 2-way task, the distribution is presented in table below:

Answer Type	**In Gold**	**Correct offered by our system**	**Total offered by our system**	**Precision**	**Recall**	**F-measure**
Yes	300	260	379	68.60%	86.67%	76.58%
No	300	181	221	81.90%	60.33%	69.48%
Total	600	441	600	73.50%		

Table 13: Results in RTE5 on 2-way task

The results are similar to those of the 3-way task and we notice the very high precision for *No* cases (81.9%), where from 221 answers offered by our system 181 are correct. The meaning of the difference between global precision from 2-way task and 3-way task is that in 31 out of 221 cases the system does no correctly distinguish between *Contradiction* and *Unknown* cases.

We can see in Table 14 that the system used for RTE-5 got comparable results with those from RTE4, with an improvement for data from the QA task. In comparison with results from RTE3 we can see that we have significant improvements on IR and IE tasks, but for the QA task, the best results were obtained for RTE-3.

Source of test data	**RTE3**	**RTE4**	**RTE5**
IR	69.00 %	82.00 %	84.0 %
QA	87.00 %	63.00 %	70.5 %
SUM	63.50 %	78.00 %	*N/A*

Source of test data	RTE3	RTE4	RTE5
IE	57.00 %	64.33 %	66.0 %
Total	69.13 %	72.10 %	73.5 %

Table 14: Comparison between results between RTE3, RTE4 and RTE5

4.2.1. Ablation Tests

In order to determine the contribution of individual components or data sources, the systems for RTE-3, RTE-4 and RTE-5 were run in turn with each component removed (Iftene and Moruz, 2010). Table 15 presents these results in parallel for RTE-3, RTE-4 and RTE-5, where the meanings for P, C and WR are: P = *Precision*, C = *Contribution* and WR = *Weighted Relevance*.

System Description	RTE-3 (69.13 %)			RTE-4 (72.1 %)			RTE-5 (73.5 %)		
	P (%)	C (%)	WR (%)	P (%)	C (%)	WR (%)	P (%)	C (%)	WR (%)
DIRT	68.76	0.37	0.54	71.40	0.7	0.97	73.33	0.17	0.23
WordNet	68.00	1.13	1.63	69.10	3.0	4.16	72.5	1.00	1.36
Acronyms	68.38	0.75	1.08	71.80	0.3	0.42	73.33	0.17	0.23
BK	67.75	1.38	2.00	70.40	1.7	2.36	72.33	1.17	1.59
NE rule	57.58	11.55	16.71	66.90	5.2	7.21	67.33	6.17	8.39
Negation rule	67.63	1.50	2.17	68.70	3.4	4.72	73.5	0.00	0.00
Contradiction rule	-	-	-	68.10	4.0	5.55	71.5	2.00	2.72
Additional processing steps	-	-	-	-	-	-	69.33	4.17	5.67
Total		16.68	24.13		18.3	25.39		14.85	20.20

Table 15: Component relevance for 2-way task

The values in each of the columns were obtained in the manner described below:

- $Precision_{Without\_Component}$ value was obtained running the RTE-3 system without a specific component (for example, $Precision_{Without\_DIRT}$ is 68.76 % and it represents the precision of the RTE-3 system without the DIRT component);
- $Contribution_{Component}$ = $Full\_system\_precision$ – $Precision_{Without\_Component}$ (for example, $Contribution_{DIRT}$ is 69.13 % - 68.76 % = 0.37 % for the DIRT component of the RTE-3 system, where 69.13 % is the precision for the full RTE-3 system and 68.76 % is the precision for RTE-3 system without DIRT component);
- $WeightedRelevance_{Component} = \frac{100 \times Contribution_{Component}}{Full\_system\_precision}$ (for example, for the DIRT component in RTE-3, $WeightedRelevance_{DIRT} = \frac{100 \times 0.37}{69.13} = 0.54\%$).

The results in Table 15 show that the system's rules related to negation, named entities and contradictions are the most important. In RTE-5 we also perform ablation tests for the module related to the additional processing steps that include preparation of input data, identification of named entities, with our patterns or using GATE, and running of TreeTagger.

Table 16 presents a comparison between ablation tests performed on RTE-5 data for the 2-way and 3-way tasks:

System Description	**2-way (73.5 %)**		**3-way (68.33 %)**	
	P (%)	**C (%)**	**P (%)**	**C (%)**
DIRT	73.33	0.17	68.00	0.33
WordNet	72.50	1.00	67.00	1.33
Acronyms	73.33	0.17	68.17	0.17
BK	72.33	1.17	66.83	1.50
NE rule	67.33	6.17	63.33	5.00
Negation rule	73.50	0.00	66.83	1.50
Contradiction rule	71.50	2.00	69.67	-1.34

System Description	2-way (73.5 %)		3-way (68.33 %)	
	P (%)	**C (%)**	**P (%)**	**C (%)**
Additional processing steps	69.33	**4.17**	64.33	**4.00**
Total		**14.85**		**12.49**

Table 16: Component relevance in RTE5

We can see in the table above the importance of the resources used for 2-way and 3-way tasks. It is interesting to see that one of the most valuable rules from RTE-4 system, the rule that identifies contradictions, has a negative contribution to the overall result of the system for the three way task.

4.2.2. Pilot Task

RTE-5 introduced a pilot task, concerning the extraction of text from a series of newspaper articles that yielded positive entailment for a given set of hypotheses. The difficulty of the task is twofold: first, the texts are not modified in any way as compared to the original source, so they may contain spelling errors, sentences with grammar errors, abbreviations and contractions, etc. The second problem is that there are a large numbers of candidate pairs, as for every one of the nine topics there are about ten hypotheses, and for every hypothesis in a topic the number of candidate pairs is equal to the number of sentences. This leads to a very large search space, and the problem to reduce it becomes very important.

In order to reduce the search space, we have made use of a technique used for our question answering systems, described in (Iftene et al., 2009a). First, using Lucene, we have indexed the articles from each topic at the sentence level, thus obtaining nine indexes. Then we have built queries for all the hypotheses by removing all punctuation and stop words, which we then used to extract the relevant text snippets. Based on experiments on the training data, we have determined that the snippets with the highest chance of yielding positive entailment are clustered around the top scoring snippets, and the first item that is not in the cluster has a Lucene score at least three times lower than that of the last item in the cluster. We have also empirically determined that the smallest feasible number of candidates is ten, and that a candidate number of above twenty is too large. In practice, the number of candidates selected is almost always above fifteen.

In order to determine the entailment value of the candidate pairs (approximately 1700 in all), we have applied a lightweight version of our entailment system. The results are given in Table 17 below:

Result	**Precision**	**Recall**	**F-measure**
Micro-average	51.12%	22.88%	31.61%
Macro-average Topic	53.03%	24.08%	33.12%
Macro-average Hypothesis	46.55%	26.42%	33.71%

Table 17: Results for RTE-5 pilot task

4.3. Results in RTE-6

The basic system described in section 5.1 was further improved for the next challenge (RTE-6), where it was entered in the main and novelty detection tasks of the RTE-6 evaluation campaign with three distinct runs, obtained by running the system with different thresholds. The results are given in table 18 below:

Run ID	**Precision**	**Recall**	**F-Measure**
001	22.89%	27.20%	24.85%
002	14.02%	39.15%	20.64%
003	31.49%	17.46%	22.46%

Table 18: Results for the RTE-6 Main Task

The first run was obtained with the thresholds set to maximize both precision and recall. Run two was obtained by lowering the threshold for separating the entailment and non-entailment cases; this is the reason for the higher recall and the lower precision. The third run was obtained by raising the value of the threshold and thus the justification for the "mirrored results".

The average, top and bottom results in the Main task are 33.72%, 48.01% and 11.60% respectively. The lower results we obtained for RTE-6 are due to the significant change in datasets; also, we did not take into account the information available in discourse, due to the lack of tools to perform such analyses (we did not include a coreference engine, for example).

The results for the novelty task are given in table 19 below:

Run ID	**Precision**	**Recall**	**F-Measure**
001	81.40%	70.00%	75.27%
002	81.54%	53.00%	64.24%
003	73.28%	85.00%	78.70%

Table 19: Results for the RTE-6 Novelty Detection Task

The runs for this task were obtained by running the system with the settings described above (in terms of thresholds); the reason for run 3 having the highest score is the difference in the scoring method. The average, top and bottom scores for this task were 77.84%, 82.91% and 43.98% respectively. The reason for which the third configuration of the system performed better for the novelty detection subtask is that most of the sentences were not novel. Because of this, by raising the entailment threshold, we reduced the number of false positives, and thus raised our recall greatly, while only slightly lowering our precision.

4.3.1. Ablation Tests

In order to determine each component's relevance, the system was run in turn with each component removed. This technique was first employed for the RTE-3 system and was subsequently used in RTE-4 and in RTE-5 (starting with RTE-5, ablation test are mandatory for all participants in the RTE challenge as they provide a good relevance score for various NLP resources). Table 20 presents these results for the system used in RTE-6, where the meanings for P, C and WR are: P = *Precision*, R = *Recall*, F = *F-measure*, C = *Contribution* and WR = *Weighted Relevance* (Contribution and Weighted relevance are computed with regard to f-measure). Ablation tests are given for our best scoring run.

System Description	RTE-6				
	P (%)	**R (%)**	**F (%)**	**C (%)**	**WR (%)**
DIRT and VerbNet	25.86	26.98	26.41	-1.56	-6.27
BK	23.91	22.01	22.92	1.93	7.76
NE rule	25.86	26.98	26.41	-1.56	-6.27
Negation rule	22.67	28.25	25.15	-0.30	-1.2
Contradiction rule	22.87	27.30	24.89	-0.04	-0.16

Table 20: Components' relevance for the RTE-6 main task

The meanings of the columns are the following:

- $Precision_{Without\_Component}$ value was obtained by running the system without a specific component (for example, $Precision_{Without\_DIRT}$ is 25.86% and it represents the precision of the system without the DIRT component);
- $Recall_{Without\_Component}$ value was obtained by running the system without a specific component (for example, $Recall_{Without\_DIRT}$ is 26.98% and it represents the precision of the system without the DIRT component);
- $F\text{-}measure_{Without\_Component}$ value was obtained by running the system without a specific component (for example, $F\text{-}measure_{Without\_DIRT}$ is 26.41% and it represents the precision of the system without the DIRT component);
- $Contribution_{Component} = Full\_system\_F\text{-}measure - F\text{-}measure_{Without\_Component}$ (for example, $Contribution_{DIRT}$ is 24.85 % - 26.41% = -1.56 % for the DIRT component of the system, where 24.85 is the f-measure for the full system and 26.41% is the f-measure for RTE-3 system without DIRT component);
- $WeightedRelevance_{Component} = \dfrac{100 \times Contribution_{Component}}{Full\_system\_f-measure}$ (for example, for the DIRT component, $WeightedRelevance_{DIRT} = \dfrac{100 \times Contribution_{DIRT}}{Full\_system\_precision} = \dfrac{100 \times -1.56}{69.13} = -6.27\%$).

Upon the examination of the output provided for the RTE-6 challenge main task, we have discovered that, due to a bug in the reporting of the results, we have mismatched some part of the results (the results reported for some of the pairs were actually extracted for other pairs). We want to emphasise the fact the entailment system works properly, and the only problem is the reporting of the results. Because the organizers of the RTE-6 challenge had not provided the gold standard for the main task at the time of writing, we were unable to check the quality of the results after the reporting bug was fixed. Because of this issue, the official results do not give a clear picture of the system's performance; also, because of the low correlation between the entailment pairs and the obtained results, the ablation tests are not relevant in the context of the RTE-6 main task. The results for the novelty extraction subtask of the RTE-6 challenge are valid.

4.4. Error Analysis

Concurrently with ablation tests, we have also carried out error analysis over the runs submitted for evaluation for the RTE-5 and RTE-6 main task data sets. Error analysis was carried out over the best scoring runs, as we made the assumption that the relevant types of errors are due to underlying conceptual problems rather than to specific parameter settings or degrees of resource coverage, and therefore will appear in any run.

In the course of the error analysis, we have found that the types of common errors differ widely from the RTE-5 data set to the RTE-6 data set. The difference is partly due to the fact that the addition of new resources and methods solved many errors, and partly due to the significant changes to the task. Below we have given a set of common errors, grouped according to type, and starting with the RTE-5 data set.

One of the most common errors we have found in the RTE-5 test run is the incorrect linking of entities (usually the linking of Named Entities) in terms of coreference. Because of this, sets of properties are not correctly transferred, and correct entailment decisions cannot be drawn. This is the case in pair 311, given, in part, below:

T: `Grey's Anatomy star Katherine Heigl and her musician hubby Josh Kelley are adopting a baby from Korea, so says the National Enquirer ...`

H: `A Korean child is going to be adopted by the "Grey's Anatomy" movie.`

The error in this case comes from the fact that we do not recognize the coreference relation between “star” and “Katherine Heigl”. Because of this, “Grey’s Anatomy” is not recognized as an attribute of the entity “Katherine Heigl”, and the system considers that the subject of the relevant sentence is “Grey’s Anatomy”; the reported result is therefore ENTAILMENT, instead of CONTRADICTION. This issue can be corrected by appropriate linking of entities. A similar problem is found for pair 337:

T: `The march was organised by Swede Annie Börjesson's mother and her best friend Maria Jansson, who between them campaigned for a fatal accident inquiry after Annie's body was found washed ashore at Prestwick in December 2005.`

H: `Maria Jansson was friends with Annie Börjesson`

The coreference relation that was not recognized is the link between “Annie Börjesson” and “her”, and because of this, the system reported UNKNOWN instead of ENTAILMENT.

Another common error is the incorrect recognition of named entities. This type of error is highlighted by pair 209, given below:

T: `Brazilian federal police arrested Jesse James Hollywood, aged 25, in Saquarema city, Rio de Janeiro, on March 8.`

H: `Jesse James was arrested in Hollywood.`

The incorrectly delimited named entity is “Jesse James Hollywood”, which is separated into “Jesse James” and “Hollywood”; since “Hollywood” is recognized as a location, the system reports ENTAILMENT instead of CONTRADICTION. The incorrect recognition of the named entity is due to the fact that we use a named entity recognizer which seems lacks robustness when encountering new names. This issue can also be seen in pair 551, where the gazetteer based NER system does not recognize “H5N1” as a named entity; because of this, the rule stating that a NE present in the hypothesis but not present in the text automatically means UNKNOWN does not fire, and the system detects a CONTRADICTION instead of UNKNOWN:

T: `Mr Lu was later found to have survived but suffered serious injuries. Mr Lu has told the Guardian that he was battered unconscious and later driven hundreds of miles to his home town where he is now recuperating. Civil rights lawyers said they were considering a legal case against his attackers, thought to be a group of thugs hired by the local authorities to put down an anti-corruption campaign against the chief of Taishi village.`

H: `Mr. Lu suffers from the H5N1 virus.`

The problem of incomplete named entity recognition can be solved in two ways: either by expanding the name database to include more names by using an external resource such as Wikipedia, or by employing a heuristic based NER system, which has greater robustness and the advantage that it is not dependent on external knowledge sources.

A less common type of error involves the incorrect analysis of predicates, in terms of argument identification, for example. The problem of incorrect treatment of predicate arguments can be observed in pair 560, where the system does not recognize the switched arguments in the hypothesis, and therefore reports ENTAILMENT instead of CONTRADICTION:

T: `Protests have continued after Manchester United defeated Lille in the Champions League round of 16.`

H: `Lille has defeated Manchester United.`

A connected problem is that the system aligns the text and the hypothesis at the syntactic relation level, and sometimes mixes relations from different sentences, without taking into account the predicate under which they appear. The problem of treating the text and hypothesis as bags of links instead of predication driven links is highlighted in pairs 302, where the system prefers the link between "Barack Obama" and "EU leaders" instead of the link to "US President". This pair can also be solved by using coreference resolution, as the entites "Barack Obama" and "US president" are in a coreference relation; since "Barrack Obama" is, therefore, a "US president", it follows that he cannot simultaneously be "an EU leader", thus proving that the solution for the current entailment pair is CONTRADICTION.

T: `Barack Obama is due to end his first overseas trip as US President today. Mr Obama has attended a series of engagements in Europe over the past week, including the G20 summit in London, a NATO meeting in Strasbourg and a conference of EU leaders in Prague.`

H: `Barack Obama is an EU leader.`

This problem can be solved by determining that both "Barack Obama" and "US president" are linked by a common predicate ("is due to end"), while "EU leaders" is not directly linked to "Barack Obama". Pair 200 is another example of incorrect predicate analysis, as the fact that the system does not make use of the semantic of the verb "defeat" leads to an incorrect result. The semantic of the verb can be analyzed by using VerbNet or FrameNet, which describe the relation between arguments:

```
T: Barack Obama made history on 4 November 2008 when he defeated
Republican rival John McCain ...
```

```
H: Barack Obama was a Republican candidate for the US presidency.
```

Some other types of errors we also found, such as incorrect global fitness calculation (pair 207), faulty application of rules (pair 459) or ignoring entities in the hypothesis (pair 443), but these are less common.

The error analysis was also carried out over the results obtained on the RTE-6 test set. The RTE-6 main task differs from any of the previous RTE main tasks in the way texts are chosen for creating entailment pairs in the sense that an RTE-6 candidate pair may contain references to information outside the entailment pair. Error analysis showed that this difference is the main source of errors in the output of our system, as can be seen below.

After examining the results of our previous system, we have found that, in order for entailment to hold, all of the named entities in the hypothesis need to be found in the text. While this is simple to test in the classical setting of the RTE challenge, in the case of the summarization setting this rule does not always work, and one of the reasons for this is gapping (information that is not explicitly present in the surface form of a text). This is a common type of error in the RTE-6 test set, as highlighted in the pair given below:

```
T: Bush made a statement in the White House Rose Garden, shortly after
Justice Sandra Day O'Connor announced her retirement.
```

```
H: Sandra Day O'Connor retired from the Supreme Court.
```

The result reported by the system is UNKNOWN, instead of ENTAILMENT because one of the named entities in the hypothesis, "Supreme Court" is absent from the text. Entailment does hold, however, because we can use information already given for the article in the title or the synopsis (usually the first sentence in a newspaper article) to augment the knowledge in each sentence. For the current example, the title of the article is "*Bush promises timely announcement on new Supreme Court Justice*" and given this extra information, it is easy to see that entailment holds. This error can also be observed in the example given below where the article title is "*Abortion issue rallies conservatives, liberals ahead of Supreme Court fight by Stephanie Griffith*":

```
T: The worst has happened with the resignation of Sandra Day OConnor,"
the group said on its website.
```

```
H: Sandra Day O'Connor retired from the Supreme Court.
```

The solution we propose for solving this type of error is that of considering certain information in any article (the title of the article could be a good approximation) as underlying any statement of that article. In practice, this would mean adding the text of the title to any candidate sentence.

Another way of referring entities outside the entailment pair is coreference. While in previous data sets this was not an issue because of the way in which those data sets were created, the lack of coreference resolution is a large source of errors. Below we give an example of such an error:

T: "This is to inform you of my decision to retire from my position as an associate justice of the Supreme Court of the United States, effective upon the nomination and confirmation of my successor," she said in a letter to Bush.

H: Sandra Day O'Connor retired from the Supreme Court.

Because the named entity "Sandra Day O'Connor" cannot be found in the candidate text, the system output is UNKNOWN, instead of ENTAILMENT. However, if the previous sentence in the document is taken into account (given below), the coreference relation between "my", "her" and "Sandra Day O'Connor" can be deduced, thus adding the missing named entity to the text. This type of error is one of the most common in the current entailment system.

P: Sandra Day O'Connor, the first woman ever appointed to the US Supreme Court, said Friday that she is retiring, giving US president George W. Bush his first opportunity to appoint a justice.

Nominal coreference is also needed, as there are cases where entailment cases are missed because entities are referred to by different names (or by parts of names), which leads the system to not align those entities and therefore make an incorrect decision. An example of such an entailment pair is given below:

T: Tyco, which has about 270,000 employees and $36 billion (euro27.5 billion) in annual revenue, makes a wide range of products including electronics, medical supplies and security devices.

H: Tyco International Ltd. has about 250,000 employees.

In this example, "Tyco" is recognized as a named entity, but since the NE in the hypothesis is "Tyco International Ltd.", the system decides that there is no viable mapping for it in the text, which leads to an UNKNOWN judgement, instead of ENTAILMENT. The solution for this type of error is to solve nominal coreference, so that the semantic relation between the two entities can be identified.

Another common source of errors in the current system is the Background Knowledge, as it does not have sufficient coverage and is overfitting for some entailment pairs. An example of BK overfitting can be seen in the example given below:

T: The court will review, in the Kentucky case, the legality of displaying framed copies of the Ten Commandments in two courthouses among copies of other documents that form the basis of US laws.

H: There is a Ten Commandments monument on the grounds of the Texas Capitol.

In the case of this entailment pair, the system first attempts to lexically match the named entity “Texas” to an entity in the text. Since this matching fails, the system resorts to alternative methods, such as using background knowledge. Since “Texas” is a part of “US”, according to BK, the system decides to align these entities in the text and hypothesis and the result for the pair is ENTAILMENT, instead of UNKNOWN. This is due to an incorrect application of the rule that states that a part of a whole can refer the whole, as this a a directional relation, and only holds if the left hand side member of the *part-of* relation is in the text and the right hand side member of the retaltion is in the hypothesis. A possible solution to this problem is to make the application of background knowledge based decisions more strictly, particularly for relations of the type “part-of” and similar relations, and to pay special attention to the directional nature of some fo the lexical-semantic relations employed by our rules.

There are also cases where the background knowledge resource does not cover all of the required cases, as can be seen in the next example:

T: The Texas case involves the legality of a stone monument inscribed with the commandments installed on the grounds of the state legislature.

H: There is a Ten Commandments monument on the grounds of the Texas Capitol.

For this pair, our system returns UNKNOWN instead of ENTAILMENT because it does not recognize any entity in the text to which the named entity “Texas Capitol” can be aligned to. However, since there is a coreference relation between “Texas” and “state”, with proper background knowledge (the legislature of an American state is a Capitol), it can be deduced that “state legislature” (or “Texas legislature” because of coreference) is equivalent to “Texas Capitol”.

Named entity recognition is a large source of errors as well, particularly in the case of less common names (such as names of books, songs, etc.), because not recognizing a named entity in the hypothesis greatly weakens its constraints over candidate texts, allowing for entailment decisions that would usually not even be considered by the system. An example of such an error is given below:

T: `Betty Friedan, a founder of the modern feminist movement in the United States, died here Saturday of congestive heart failure, feminist leaders announced.`

H: `Betty Friedan is the author of "The Feminine Mystique."`

In the case of this entailment pair, the error is derived from the analysis of the hypothesis, as the NER system does not recognize “The Feminine Mystique” as a name. Since the only recognized entity in the hypothesis is “Betty Friedan”, the fact that the candidate text has no mention of the book title just reduces the fitness score instead of rejecting the pair outright. This type of error seems to be common, as it leads to 7 incorrect ENTAILMENT results for the above hypothesis alone, a very large error margin for one hypothesis. On average, a hypothesis may have up to 100 candidate texts, of which about 5% are entailments (Bentivogli et al. 2011). This problem can be addressed by using a heuristic based RTE engine that can compensate for incomplete knowledge, as it is unfeasible to create, manage and use a sufficiently comprehensive gazetteer. A simple heuristic that would solve this issue is based on the fact that the book title in this example is sorrouded by inverted commas; if all the words between the inverted commas, apart from prepositions and conjuctions, are capitalezed, than the text delimited by the iverted commas is a named entity.

We have also identified errors due to the thresholds used to separate the ENTAILMENT cases from the NO ENTAILMENT cases, which means that it may be more difficult to separate them in the context of recognizing textual entailment in a summarization setting.

The error analysis described in this section leads us to believe that improving named entity recognition and adding a tool for extracting corefering entities are the best ways to improve our result. One of the main error sources for the system used in RTE-5, lack of predicate analysis, is no longer an issue for the improved version of the entailment engine that was used for RTE-6.

4.5. Conclusions

In this section we have given a brief overview of the entailment system developed for the RTE-5 (Iftene and Moruz, 2010) and RTE-6 (Iftene and Moruz, 2011) challenges. The system is based on that of (Iftene, 2008), to which new resources and rules have been added, as described in (Moruz, 2010). The latest version of the system, which was used for the RTE-6 challenge, is further enhanced by the use of the VerbNet lexical-semantic database, which, itself, is used for detecting verb correlation; many of the resources feeding the initial system are kept as well, as they provide significant amounts of lexical and background knowledge. To compensate for lack of coverage, some resources are used in conjunction (VerbNet and DIRT, for example).

In terms of results, the system described above performs well, as proved by the results in the RTE-5 challenge, where it was ranked first in both main tasks, and fourth in the pilot task. For the RTE-6 challenge, the results reported to the organizers, on the basis of which the system was ranked, were not correlated properly to the results given by the system, because of a bug in result reporting; however, the system performed well on the novelty detection subtask, and was ranked fifth.

Given the good results that the system obtained on the RTE-5 data set, we conclude that the system has reached a certain maturity in solving the original formulation of the Textual Entailment task. The innovations we have proposed proved adequate, advancing the state of the art in the field, and moving the solving of textual entailment towards a deeper semantic analysis. Nevertheless, we believe that there is still room for significant improvement until a consistent and elegant solution can be provided for the problem of recognizing textual entailment. It is our sincere opinion that significant progress in textual entailment will be obtained when the issue of deep semantic understanding of natural language will be properly addressed. This conclusion is also supported by the ablation tests, which show that most knowledge sources we have used in the RTE-5 challenge greatly increased performance; the largest gain comes from better preprocessing, which increases both the performance of the rule application and the performance of the preprocessing tools.

As for the summarization setting of the textual entailment challenge, we have found that better preprocessing of the texts (more careful management of named entities, for example), coupled with the addition of some tools for discourse analysis, such as a coreference resolver, would significantly improve the result.

5. Romanian Lexical-Semantic Resources for Entailment

As we have discussed in chapter 3, the basis of the approach proposed for solving textual entailment is having access to lexical semantic resources such as VerbNet, WordNet, SUMO-MILO, etc. While this allows for great flexibility in the approach in terms of adapting it to another language, it does require that the basic resources be available for the language in question. Consequently, if we want to adapt our entailment system for the Romanian language, we would require Romanian versions of VerbNet and WordNet (VN is used for predicate matching, while WN is the lexical ontology required for argument matching) at the least. Since most notable lexical-semantic resources for the English language are already aligned to WordNet, it stands to reason that having a Romanian version of WordNet that is aligned at the synset level to the English WordNet would greatly help in developing these resources. This chapter discusses ways in which existing lexical-semantic resources for the Romanian language can be enhanced by using the electronic format of the Romanian Language Thesaurus Dictionary (eDTLR), thus allowing for the adaptation of the algorithm in section 3 for Romanian.

5.1. The Romanian WordNet

The Romanian WordNet (Tufis et al., 2004) was built as part of the BalkaNet multilingual lexical semantic network (Stamou et al., 2002). BalkaNet exploits the concept of the inter-lingual index (ILI), introduced by the EuroWN project (Rodriguez et al., 1998). The ILI structure can be viewed as a semantic network, whose nodes are language independent concepts that are linked by labelled directed arcs that represent the semantic connections between concepts. In order for the Romanian version of WordNet to be representative to the Romanian language, the authors considered that the best approach to its creation would be a language centric one, as opposed to a translation of the synsets of the Princeton WordNet. To this end, the authors relied on Romanian lexicographic resources such as the Explanatory Dictionary of Romanian, The Dictionary of Synonyms, as well as an in-house Romanian-English dictionary. In terms of coverage, the current version of the Romanian WordNet (Tufis, 2008)(Tufiş and Ştefănescu, 2011) covers all the Base Concepts in EuroWordNet, concepts commonly implemented in most EuroWN WordNets, a series of concepts proposed by the partners of the BalkaNet project, as well as Romanian specific concepts, thus offering significant cross-lingual coverage. The terms proposed for the Romanian language are extracted according to frequency in a newspaper corpus, the number of senses for the word, and its definitional productiveness (i.e. the number of sense definitions a word is used in). The set of ILI candidates that might represent a given word was extracted based on the in-house bilingual dictionary, and the final alignment was then chosen by lexicographers.

After implementing the ILI concepts for Romanian, the authors examined the nature of the semantic relations between them. Even though some of those semantic relations are language dependent (such as derivative or direct antonymy, for example), many relations can be automatically imported into Romanian. Table 21 below gives the relations that have been automatically mapped.

Relation	Valid POSes for the relation	Imported
hypernym	<N, N>; <V, V>	yes
holo_part	<N, N>	yes
holo_portion	<N, N>	yes
holo_member	<N, N>	yes
subevent	<V, V>	yes
causes	<V, V>	yes
verb_group	<V, V>	yes
be_in_state	<A, N>	yes
similar_to	<A, A>	yes
also_see	<V, V>; <A, A>	yes
category_domain	<N, N>; <V, N>; <A, N>; <B, N>	yes
near_antonym	<N, N>; <V, V>; <A, A>; <B, B>	yes but with restrictions
derived	<A, A>; <B, A>; <A, N>	partially

Table 21: Relations in PWN 2.0 that are subject to import in the Romanian WN

The description of each of the relations in Table 12 is given below, along with examples from the English WN, taken from (Tufiş et al., 2004):

- *Hyperonymy*, denoted by hypernym, is a semantic relation which establishes a specific-generic relationship between the related meanings;
- *holo_part* denotes a PART-OF relation in PWN 2.0: (finger:1)→(hand:1);
- *holo_portion* denotes a SUBSTANCE-OF relation in PWN2.0: (wood:1)→(timber:1);
- *holo_member* denotes a MEMBER-OF relation in PWN2.0: (tree:1)→(forest:1)
- *subevent* denotes that the activity described by one argument is temporally included in the activity denoted the by the other: (snore:1)→(sleep:1);
- *causes* denotes that the verb concepts are in a causative relation:(kill:1)→(die:1);
- *verb_group* groups several similar overlapping meanings of the verbs: (act:2 behave:1 do:9)→((act:5 play:8 act-as:2)(dissemble:3 pretend:2 act:9))
- *be_in_state* specifies a value for a property. The values related by *be_in_state* are represented by descriptive adjectival synsets and the properties by nominal synsets: (tall:1)→(stature:2 height:3).
- *similar_to* denotes a relation between adjectival meanings analogous to the *verb_group* relation for verbal meanings
- *also_see* links semantically related verbs and similar adjectival clusters: ((tall:1 vs. short:3))→(((high:2) vs. (low:2))((large:1 big:1) vs. (small:1 little:1))).
- *category_domain* denotes topical classifications of the meanings represented by the synsets; the target synset of the relation is always a nominal synset, but any synset, irrespective of its POS can be source of this relation: (diplomatic immunity:1)→(law:2 jurisprudence:2)
- *near_antonym*. Antonymy is not granted for fully automatic import, so it is manually checked
- *derived* denotes a lexical relation that links derivatives to their stems (quickly→quick), (astomatal→stomatal) (abbatical→abbey).

The Romanian version of WordNet described in (Tufiş et al., 2004) contains 16566 synsets, and approximately 30000 token literals (the statistic is given for the 17th May 2004 version). Tables 13 and 14 below show the coverage of the reported version of WordNet. Even though the coverage offered is good, it is insufficient compared to the over 115000 synsets in the Princeton WordNet; also, real life applications, such as textual entailment, require greater coverage. To this extent, and based on our experience with dictionary parsing, we present an idea of automatically expanding the Romanian WordNet by using the electronic format of the Romanian Thesaurus Dictionary (eDTLR) (Cristea et al, 2007) and a series of simple heuristics. It is important to note that such an extension would not be as accurate (in terms of sysnset or hierarchy quality, for example) as the manually corrected version described in this subsection, but, for the purposes of our approach, even a less than perfect extension would be greatly useful.

Noun synsets	Verb synsets	Adj.synsets	Adv. synsets	Total
10725	4164	844	833	16566

Table 22: POS distribution of the sysnsets for the 2004 version of RoWN

Language	Synsets	Token literals	Type literals	Average synset length	Average senses/lit
Romanian	16566	29130	17538	1,75	1,66
English	115424	203147	145627	1,76	1,39

Table 23: Comparison between the 2004 RoWN and PWN

Since the 2004 version of the Romanian WordNet, it has been further extended by adding more literals and synsets. At the moment of writing, the authors report the statistics shown in table 24 below[20]. Unfortunately, since this latest version ROWN is unavailable to us, we had to make use of the latest available version, which is the one described above. Examples of RoWN synsets are given in annex 3.

	Noun	Verb	Adjective	Adverb	Total
Synsets	40540	10044	4968	3173	58725
Literals	38490	6766	4251	2850	52357

[20] Data taken from http://www.racai.ro/wnbrowser/Help.aspx

Senses	55861	16402	8754	4158	85175

Table 24: Statistics for the current Romanian WordNet (at the time of writing)

5.2. The Electronic Format of the Romanian Thesaurus Dictionary (eDTLR)

The need for machine readable dictionaries has been apparent for years now, and a number of attempts have been made to transform various printed dictionaries into an electronic form (the most relevant example is that of the Trésor de la Langue Française Informatisé[21]) Electronic dictionaries in text format, however, are not very useful, and so the need arises for a structured version of for dictionary entries that can be easily searched. The parsing of a dictionary entry consists of the creation of a lexical-semantic tree of senses corresponding to the meanings that define the dictionary lexical entry. In (Curteanu et al, 2008b) the authors introduce a new parsing strategy for dictionary shallow parsing, called *Dictionary Sense Segmentation & Dependency* (DSSD), for extracting the *sense tree* (the hierarchy of the lexical-semantic meanings) for a dictionary entry. The DSSD algorithm was first used for parsing the entries of the Romanian Language Thesaurus (DTLR – Dicţionarul Tezaur al Limbii Române) within the eDTLR research project[22] (Cristea et al., 2007), (Cristea, Raschip and Moruz, 2009). Our contribution to this project was significant, and includes, among others, the creation of the parsing algorithms, the analysis and creation of the sense marker hierarchies (together with N. Curteanu and D. Trandabăţ), the analysis of sense definitions (also with N. Curteanu and D. Trandabăţ) and the implementation and testing of the parser.

Given a dictionary entry, sense trees are obtained in two steps: first, the primary and secondary senses have to be identified by means of typographical markers, and then, the identified senses are arranged in a sense tree on the basis of a predefined hierarchy. For the DTLR dictionary, the sense markers hierarchy includes 6 levels. The DTLR sense markers are, from the most general to the most specific: *capital letter* markers (**A.**, **B.**, etc.), *Roman numeral* markers (**I.**, **II.**, etc.), *Arabic numeral* markers (**1.**, **2.**, etc.), *filled diamond* (♦), *empty diamond* (◊) and bold and italic definition markers (used for expressions and idioms). Apart from the six marker types described above, there also exists a special type of sense marker that does not have

[21] http://atilf.atilf.fr/

[22] The present research was partly financed within the eDTLR grant, PNCDI II Project No. 91_013/18.09.2007.

a fixed position in the hierarchy, the *literal enumeration*, consisting of *lowercase letter* markers (**a)**, **b)**, **c)**, etc.). The literal enumeration can be attached to any of the existing six levels, as a means of refining sense description. Therefore, on the basis of sense markers, any dictionary entry is represented as a tree of senses, the lower levels being more specific instances of the higher levels.

For example, for the dictionary entry *verb*, the sense tree contains three senses corresponding to level 3, one of them having a sub-sense corresponding to level 5. Each sense can have its own definition or examples. Also, all definitions can be modified using reserved words, thus changing the scope of any given gloss.

```
<entry>
  <hw>VERB</hw>
  <senses>
    <marker level=”3”>1.
      <definition>…</definition>
      <marker level=”5”>◊
      <definition>…</definition>
      </marker>
    </marker>
    <marker level=”3”>2.
      <definition>…</definition>
    </marker>
    <marker level=”3”>3.
      <definition>…</definition>
    </marker>
  </senses>
</entry>
```

The method described for the parsing of the DTLR thesaurus is robust, and can be adapted to any dictionary, provided that the general semantic structure of the dictionary entries is known and the sense hierarchy is defined.

5.2.1. Dictionary Sense Segmentation and Dependency (DSSD)

As described in (Curteanu et al., 2008b), the DSSD parsing strategy is based on the idea of the SCD (Segmentation-Cohesion-Dependency) parsing strategy, developed and applied to Romanian free text analysis and described at large in (Curteanu, 2006). DSSD is extended on the basis of the similarity between the SCD theory and dictionary entry parsing: SCD is based on the notions of discourse marker classes that are arranged in a hierarchy, while DSSD relies on sense marker classes, also arranged in a hierarchy. The data structures obtained after parsing are also similar, as SCD creates discourse trees with finite clauses in the nodes, and DSSD builds sense trees, with sense definitions in the nodes. Despite the formal similarity between the data structures, each type of tree models different relations, as the nucleus-satellite relation described by the SCD tree is completely different semantically from the lexical-semantic based subsumption relation described by sense trees. This is the reason for the DSSD derivation of SCD to keep the dependency and segmentation aspects of the theory, and to discard the cohesion aspects, as they have no meaning in the context of dictionary entry sense trees. The lexical subsumption relation is defined as follows: $sense_1$ subsumes $sense_2$ if $sense_1$ is less informative (or, more general) than $sense_2$; in terms of dictionary entry structure, $sense_1$ subsumes $sense_2$ if the sense tree of $sense_2$ is a subtree of $sense_1$.

While the SCD and DSSD parsing strategies seem quite different, as SCD is devoted to free text parsing and DSSD is used for dictionary entry parsing, they are however related. The two strategies work formally with the same technology, using similar analysis tools and data structures, including the same Breadth-First search strategy. The significant distinction between the two comes from the very different kinds of texts that each has to analyze (free text and dictionary entry text), and the two different (but complementary) semantics that drive the corresponding parsing structures: predicational and rhetorical (cohesion-proper) semantics for SCD, and lexical semantics (cohesion-free) for DSSD (Curteanu et al., 2008).

The proposed method analyses dictionary entries using two SCD configurations: one for building a tree of senses and subsenses for each dictionary entry (usually a word), and another SCD configuration for identifying, for each node of the sense tree (corresponding to a specific meaning of the entry word), the types of definitions described (i.e. regular definition, usage specifications, usage examples, etc.).

An SCD configuration has the following components:

- A set of marker classes: a marker is a boundary for a specific linguistic category.
- A hierarchy that establishes the dependencies between the marker classes.
- A parsing algorithm.

The parsing algorithm is designed to perform the following actions: recognize markers, thus identifying the structures they bound, and classify these structures according to the hierarchy. The algorithm can be applied to different marker classes or hierarchies, strictly depending on the semantics of the text to be parsed.

5.2.2. Advantages of DSSD over Standard Dictionary Entry Parsing (DEP)

This section outlines the novelties of the DSSD approach compared to the standard DEP, *e.g.* (Neff and Boguraev; 1989), (Lemnitzer and Kunze; 2005), (Kammerer, 2000). DSSD applies the same "technology" as the SCD strategy does, *i.e. marker classes*, *specific hierarchies*, and adequate *searching procedures* which govern the parsing algorithms. Most importantly, DSSD parses and builds the sense tree of a (DTLR) dictionary entry, independently of sense definition parsing process. In the standard DEP, building the sense tree for an entry is inherently embedded into the general process of parsing all the sense definitions enclosed into the dictionary entry. This is the case for the dictionary entry parser employed by (Neff and Boguraev; 1989) or *LexParse*, (Kammerer; 2000: 10-11) specifying that the *LexParse* recognition strategy is a *Depth-First*, *Top-Down* one.

The advantage of the proposed DSSD approach is that it does not parse, at least in the beginning, the details of sense definitions; the first step of the DSSD parsing is concerned with discovering the sense markers in a dictionary entry and establishing dependencies between them. The second step of dictionary entry parsing, definition analysis, is then carried out over a already formed sense tree, thus greatly reducing the error rate of both steps, and also guaranteeing that the result of the parsing process is always a valid sense tree (in the case of standard DEP, if the parser encounters an error, the entire sense tree is discarded).

Based on different types of DTD standards for dictionary text representation, such as CONCEDE-TEI (Erjavec et al. 2000; Tufis 2001) or (XCES-TEI; 2007), the parsing process may continue "in depth" for identifying sense definitions. The DSSD strategy has the quality of being able to independently compute the entry sense tree, prior to the process of sense definition parsing. Subsequently, the process of parsing the sense definitions can be performed separately and sequentially, avoiding the frequent situation when the general parsing of an entry may be stopped simply because of an unparsable sense definition (even if this unparsable definition is the last one) (Curteanu et al., 2008b).

The procedural *pseudo-code* in Fig. 3 clearly shows the important difference between *standard* DEP and DSSD *parsing*, with the essential advantage provided by DSSD: standard DEP is based on *Depth-First* search, while DSSD works with *Breadth-First* search. Specifically, the procedural running of the four operations that are compared for the standard DEP and DSSD strategies, labeled with ①, ②, ③, ④, are organized in quite different cycles: in the table left-side (standard DEP), there is a single, large running cycle, ① + ②, under ② being embedded (and strictly depending) the sub-cycle ③ + ④. The DSSD parsing exhibits two distinct (and independently) running cycles: ① + ④, for constructing the (DTLR) sense trees, and ② + ③, devoted to parse the sense definitions and to attach the parsed or unparsed sense definitions to their corresponding nodes in the sense tree(s).

Dictionary Classical Parsing Strategy	**DSSD Parsing Strategy**
For *i* from 0 to *MarkerNumber* ① **Sense-*i* Marker Recognition;** ② **Sense-*i* Definition Parsing;** *If*(Success) ③ **Attach (Parsed) Sense-*i* Definition to Node-*i*;** ④ **Add Node-*i*** to **EntrySenseTree;** *Else* Fail and Stop. EndFor **Output: EntrySenseTree** *with* **Parsed Sense Definitions** (only if all sense definitions are parsed). **Notice:** *MarkerNumber* is the number of the input marker sequence.	For *i* from 0 to *MarkerNumber* ① **Sense-*i* Marker Recognition;** **Assign (Unparsed) Sense-*i* Definition** to **Node-*i*;** ④ **Add Node-*i*** to **EntrySenseTree;** **Standby** on **Sense-*i* Definition Parsing;** EndFor **Output: EntrySenseTree.** **Node-*k* =** Root(**EntrySenseTree**); *While* not all nodes in **EntrySenseTree** are visited ② **Sense-*k* Definition Parsing;** *If*(Success) ③ **Attach Sense-*k* Definition to Node-*k*;** *Else* **Attach Sense-*k* Parsing Result to Node-*k*;** **Node-*k* =** getNextDepthFirstNode(**EntrySenseTree**) Continue EndWhile. **Output: EntrySenseTree** *with* **Parsed** *or* **Unparsed Sense Definitions**

Figure 3: A pseudo-code comparison of the classical and DSSD dictionary parsing strategies

The second procedural cycle is optional, and that the first cycle is working on the sense marker sequence of the entry (either correct or not), the DSSD output being an entry sense tree in either case (either correct or not). This is why the DSSD algorithm never returns FAIL, regardless whether the obtained sense tree is correct or not. In the case of errors, it leaves spans of input text unparsed but gets back on the correct track each time a non-ambiguous marker is found. Figure 1 was initially presented in (Curteanu et al., 2008b), a paper to which we also brought significant contributions.

5.2.3. DTLR Marker Classes, their Dependency Structure, and the DSSD Parsing Algorithm

As already pointed out, DSSD can be viewed as a simplified version of SCD, since only the *segmentation* and *dependency* aspects are involved, the (local) *cohesion* matters being without object for the (one-word) lexical semantics of DSSD. As in the case of SCD, the DSSD parsing strategy requires a set of *marker classes* (in our case, DTLR sense markers), arranged in a *hierarchy* illustrated in Fig. 2, and described below:

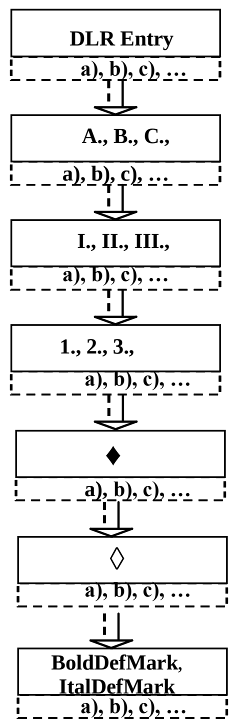

Figure 4: Hierarchy of DTRL sense markers

The *capital letter* marker class (**A.**, **B.**, etc.) is the topmost level on the sense hierarchy of DTLR markers for any given dictionary entry. When it appears, this marker designates the (largest-grained meaning) *primary senses* of the lexical word defined. If the top level marker has only one element of this kind, then the marker is not explicitly represented.

The *Roman numeral* marker class (**I.**, **II.**, etc.) is the *second-level* of sense analysis for a given DTLR entry. It is subsumed by a capital letter marker if some exists for the head word; if a capital letter marker does not exist (it is not explicitly represented), the Roman numeral marker appears on the topmost level of the sense tree. If the lexical entry has only one sense value for this analysis level, the marker is not explicitly represented.

The *Arabic numeral* marker class (**1.**, **2.**, etc.) is the *third-level* of sense analysis for a DTLR entry. It is subsumed by a Roman numeral marker if there exists some for the entry; if a Roman numeral marker is not explicitly represented, it is subsumed by the first explicit marker on a higher level. If the entry has only one sense value for this level of sense analysis, the marker is not explicitly represented. These first *three levels* encode the *primary senses* of a DTLR lexical entry.

The *filled diamond* marker class is the *fourth-level* of sense analysis and it is used for enumerating *secondary* (finer-grained) *senses* of a DTLR entry. It is generally subsumed by any explicit DTLR sense marker on a higher level, *i.e.* any of the primary sense markers.

The *empty diamond* marker class is the *fifth-level* of sense analysis and it is used for enumerating expressions for a given, *secondary sub-sense*. It is generally subsumed by a filled diamond marker or by any primary sense marker.

The *BoldDefMark / ItalDefMark* marker class is the *sixth-level* of sense analysis and it is used for giving idioms and expressions that contain the current sense of the title word, and constitute a *secondary sub-sense*. It is generally subsumed by an empty diamond marker or by any primary sense marker.

The *lowercase letter* markers **a)**, **b)**, **c)**, etc. are not an actual class of sense markers, but rather a *procedure* used to refine, through *literal enumeration*, a semantic paradigm of a DTLR entry sense or sub-sense. A lowercase letter marker does not have a specific level on the marker class tree-like hierarchy since it belongs to the sense marker level (of either primary or secondary sense) that is its parent. The important rules of the *literal enumeration* procedure in DTLR are: **(a)** it associates with the hierarchy level of the sense marker class to which is assigned (in Fig. 3), and **(b)** it can embed lower (than its parent level) senses, provided that each literal enumeration is closed finally on the sense level to which it belongs.

Fig. 4 shows the hierarchy of the DTLR sense marker classes. The *continuous-dashed arrows* in Fig. 4 point downwards from the higher to the lower priority levels of DTLR marker class hyper-tree. Because of its special representation characteristics, the literal enumeration is illustrated on a layer *attached* to the hierarchy level to which it belongs, on each of the sense levels.

The DSSD algorithm for the construction of the DTLR sense tree, according to the marker hierarchy described in Fig. 4, is the following:

```
Stack S
Tree T
S.push(root)
while article has more markers
  crt = get_next_marker()
  while crt > S.top() – get to the first higher rank marker in the
stack
    S.pop()
  if(crt = lowercaseLetter)
    S.top.addPart(crt) – add a lowercase marker as a subset of the
higher level sense value
    crt.level=S.top.level+1 – the lowercase letter maker is given
a level in accordance to the level of its parent
    S.push(crt)
  else
    S.top.add_son(crt) – add the son to the higher level marker in
the stack
    S.push(crt) – add the current marker to the stack
```

While the DTLR sense marker recognition in DSSD is achieved with a *Breadth-First* search, the marker sequence analysis for sense tree construction is based on a *Depth-First* parsing of the sense marker sequence input, as it uses a stack to keep track of previous unfinished (in terms of attaching subsenses) sense markers.

5.2.4. DSSD Parser Applied on DTLR Entries

Fig. 5 shows the result of applying the parser described previously on a DTLR entry. We notice that the presented input example (*VENIT*[2]) represents just sequences of DTLR sense markers. The entry for which the parsing was conducted is given only in part (we have only shown the aspects relevant to the sense tree, as the entire entry spans for more than two dictionary pages):

```
– <entry>
    <list>VENIT2, -Ă 1. 2. ◊ a) b) c) ◊ a) b) n-11</list>
  – <node value="VENIT2, -Ă" class="0">
      <node value="1." class="6"> </node>
    – <node value="2." class="6">
      – <node value="◊" class="10">
        – <parts>
            <node value="a)" class="11"> </node>
            <node value="b)" class="11"> </node>
            <node value="c)" class="11"> </node>
          </parts>
        </node>
      – <node value="◊" class="10">
        – <parts>
            <node value="a)" class="11"> </node>
            <node value="b)" class="11"> </node>
          </parts>
        </node>
      </node>
    </node>
  </entry>
– <entry>
```

Figure 5: Sense tree for the eDTLR entry *VENIT*^{2}

As one can see, the input of the sense tree parser is the DSSD marker sequence of the considered DTLR entry (the <list> tag in Figure 3). The output of the parsing is much less verbose than the original dictionary entry, since the sense definitions and the entire example text is not depicted, in order to better observe the sense tree of the entry. Also, this representation proves that the understanding of the sense definitions is not strictly necessary for building the sense tree, a task for which the marker hierarchy discussed in the previous section is sufficient.

Analysis of the semantic content of DTLR senses is carried out by a second SCD configuration, which is based on a set of definition markers (Curteanu et al 2008b). The markers are used to create a finite number of templates which are then used for extracting subsenses. DTLR entries have the following types of definitions:

- *MorfDefs* – morphological definitions, which give the part of speech of the current sense;
- *RegDefs* – definitions written in regular font, that are, in most cases, glosses for the current sense;

- *BoldDefs* – definitions written in bold. These definitions are in fact sense markers, and are used for introducing an expression;
- *ItalDefs* – definitions written in italics. These definitions are in fact sense markers, and are used for introducing an idiom;
- *SpecDefs* – specifications on the current sense (for example usage information);
- *SpSpecDefs* – specifications on the current sense regarding specific information, such as grammatical knowledge (transitivity for verbs);
- *DefExems* – examples, given as explanations for the current sense.

The definition types proposed here receive specific functional roles in describing the meanings under primary and secondary senses (Curteanu et al.; 2008b). Two taxonomies of DTLR definitions can be defined. The first contains the following classes:

- **(obli)** *compulsory definitions*, which are the *MorfDef*s and, for each DTLR entry, *one* of the following three definitions: either *RegDef*, or *BoldDef*, or *ItalDef*. The meaning of obligatory definitions is that there are no entries that have no *MorfDef* and (at least) *one* of the definitions from the set {*RegDef*, *BoldDef*, *ItalDef*}.
- **(opti)** *optional definitions* in DTLR: *SpecDef*s, *SpSpecDef*s, and *DefExem*s, whose presence is optional as modifiers for an obligatory definition.

The other taxonomy separates definitions in:

- **(auto)** *autonomous definitions*: *RegDef*, *BoldDef*, and *ItalDef*, meaning that these definitions have a stand-alone role in introducing DTLR senses;
- **(cont)** *contingent definitions*: *MorfDef*s, *SpecDef*s, *SpSpecDef*s, and *DefExem*s, which have no standalone meaning.

MorfDef is compulsory at the root level of any DTLR entry, being inherited on the lower levels of the sense tree. *MorfDef* is both a compulsory (at the root level) and a *contingent* definition, being placed in front of an *autonomous* definition (usually, a *RegDef* one). *SpecDef*s, *SpSpecDef*s, and *DefExem*s are *contingent* definitions since they cannot define a subsense in an autonomous manner but only as auxiliary methods that modify either autonomous definitions or other contingent definitions.

Morphological Definitions – *MorfDef*s

The *morphological definitions* (*MorfDef*s) consist of one or more labels describing morphological or phrasal categories, at various levels of the sense tree. The very first element in a dictionary entry is a *MorfDef* detailing all the possible morphological categories that the current word may belong to. As senses become more refined, subsequent *MorfDefs* become more and more specific up to the point where they designate a single morphological category. Those DTLR senses that lack a *MorfDef* have to inherit it from the closest upward regent sense where such a *MorfDef* is lexically present. In what follows, examples of all definitions will be highlighted in grey.

VERZIŞÓR, -OÁRĂ adj., subst. **I.** Adj. Diminutiv al lui v e r d e (**I 1**)... **II.** Subst. **1.** S. M. (La pl.) Corp de trupă al cavaleriei... **2.** S. M. Şi f. (Iht.; prin Munt.) Boiştean... **3.** S. N. (Prin Mold.; în forma *verdişor*) Rachiu cu mentă... **4.** S. F. (Regional) Varietate de struguri... **5.** S. N. (Familiar) Bancnotă de culoare verde...

Regular-font Definitions – *RegDef*s

Regular definitions (*RegDef*s) are the most common linguistic instrument used in DTLR to describe the senses. A *RegDef* is a span of text in *regular font* which is not fully enclosed within brackets (but may contain bracketed text), and with the exception of several reserved words and abbreviations. *RegDef* represents the gloss of a word, as it is used in most common dictionaries.

VENÍRE s. F. Acţiunea de a v e n i şi rezultatul ei. **I. 1.** Deplasare către cineva sau către ceva; parcurgere a unui traseu pentru a ajunge la un anumit loc,...

◊ E x p r. **Bun venit =** formulă de salut prin care se exprimă mulţumirea în legătură cu sosirea, cu prezenţa cuiva.

◊ *Venit naţional* = parte a produsului economiei naţionale dintr-o perioadă de timp, care rămîne după...

RegDef, possibly accompanied by other contingent definitions, together with the lexical entry root or a sense marker constitutes the complete description of that sense. *RegDef* can also occur refined by literal enumeration as shown below:

VIOÁRĂ[2] s. F. (Regional, mai ales în Transilv.) Numele mai multor plante erbacee: **a)** toporaş **(II 1 a)** (*Viola odorata*). ...; **b)** plantă din familia cruciferelor, cu tulpina simplă sau ramificată, cu frunzele mici, rotunde-ovale, crestate, de culoare verde-deschis, acoperite cu peri,...

Bold-font Definitions – *BoldDef*s

A *BoldDef* definition is devoted to explaining the meaning of a specific *phrase* or *utterance*, its general form being a *bolded pattern* of that phrase, followed by a *BoldDef* separator (typically "="[23]), and followed by a *RegDef* definition. Usually, *BoldDef*s refine more specific subsenses as compared to the DTLR secondary senses introduced by ♦ and ◊.

◊ **A semăna în verde** = a semăna imediat după arat, cînd arătura este încă proaspătă. … **A ara în verde =** a ara un pămînt care este încă jilav. ...

There are situations when *BoldDef*s may occur in primary senses, including at the *root level* of a DTLR lexical entry. A *BoldDef* can be very complex, containing different variants of the bolded expression.

3. A se duce (sau a merge, a se lăţi, învechit şi regional, a ieşi) vestea (cuiva, a ceva, de ceva etc.) sau a i se duce (ori a-i merge, a i se lăţi, învechit, rar, a i se ridica, regional, a-i ieşi cuiva) vestea, a-i merge (sau a i se duce cuiva) vestea şi povestea, (învechit şi regional) a ieşi veste (de cineva sau de ceva) = a 115ocial foarte bine cunoscut, a i se duce faima;...,

Ital-font Definitions – *ItalDef*s

ItalDefs are syntactically similar to *BoldDefs*, but are semantically different, as they usually describe collocations, as compared to *BoldDefs* which describe expressions.

Verde antic = matostat. ...

VERZÉR subst. (Regional; în sintagma) *Verzerul tilegii* = schimbătoare la roţile plugului...

Specification-based Definitions – *SpecDef*s

SpecDefs are contingent definitions written in regular font and enclosed into brackets. Most of them are abbreviations or reserved words denoting various usage contexts, as for instance: "(Regional usage)", "(Slang)", etc. We point out that sometimes *SpecDefs* do not occur

[23] Sometimes the "=" separator is replaced by equivalent expressions, such as "vezi", "v.", "se spune", etc., and introduces a relation meaning that the expression in bold on the left hand side is semantically equal to the right hand side member

within brackets, but this is only so when they are reserved words or abbreviations. *SpecDefs* are used at any sense level and modify definitions, as shown in the examples below.

(1) In the entry root, immediately after *MorfDef*:

VENIÁL, -Ă adj. (Livresc; despre păcate[2], greşeli etc.) Care poate fi iertat (de Biserică); uşor, fără importanţă…

(2) *SpecDef*s in the root of a primary sense, (without or with literal enumeration):

2. (Învechit şi regional; despre lichide, substanţe etc.) Veninos **(2)**. ...

(3) *SpecDef*s in the root of a secondary sense, (without or with literal enumeration):

♦ F i g. (Despre oameni) Rău **(A I 1)**; duşmănos; (despre manifestări, stări, acţiuni etc. Ale oamenilor) care trădează răutate **(I 1)**,...

Spaced-character Definitions – *SpSpecDef*s

Another contingent definition is *SpSpecDef*, which stipulates various standard features; it is written with spaced-literals from an established list of abbreviations, *e.g.*:

SpSpecDef may occur at all DTLR sense levels, sometimes together with other contingent definitions, as in the examples below:

2. T r a n z. Şi r e f l. F i g. A (se) amărî, a (se) supăra, a (se) necăji, a (se) mînia. *A sa prea iubită inimă ş-a veninat.* PANN, E. II, 94/18.

Some cross-references are written with spaced font, thus one must take care to always check whether a given spaced word is part of the list of *SpSpecDefs* or not.

Plantă erbacee din familia scrofulariacee, cu florile albe sau trandafirii, care creşte în locuri umede sau mlăştinoase şi care este folosită în medicină pentru proprietăţile ei iritante şi purgative; avrămeasă, (regional) milostivă (v. M i l o s t i v **III 2**), potroacă1 (4), mila-Domnului (v. M i l ă1 **I 6**) (Gratiola officinalis). Cf. Hem 2182, conv. Lit. Xxiii, 1060, brandza, fl. 349, 116oci, t. 188, barcianu, jahresber. Viii, 101,...

Definition-meanings by Examples – *DefExem*s

Finally, the *autonomous* definitions may receive, each of them in case of multiple occurrence, one or several *examples*, from bibliographic sources referred by *sigles* or created by the dictionary authors, refining the meanings of each autonomous definition. A sequence of *DefExems*, each followed by a *sigle* is given below:

A intra în viaţă = **a)** (despre oameni; şi în forma **a păşi în viaţă**) a începe să se confrunte cu realitatea. *Cum a intrat el în viaţă? Cît amor de drept şi bine, Cîtă sinceră frăţie adusese el cu sine?* EMINESCU, O. I, 53. …; **b)** (rar) a începe să activeze, să funcţioneze. *Guvernul cel nou... va intra în luna lui marţiu în viaţă.* VASICI, ap. BARIŢIU, C. II, 47.

Special Observations and Situations

(1) The procedure of *literal enumeration* is frequently met in DTLR, and can be applied to each autonomous definition, independently on the sense level which this definition occurs on.

(2) In DTLR one may encounter (non-standard) shapes that match none of the already discussed definitions. The following DTLR entry is neither *RegDef*, or *BoldDef*, or *ItalDef*.

VIDUÍ vb. IV. R e f l. (Prin Olt.) = zvidui. Cf. VÎRCOL, V. ...

5.2.5. Result Analysis

Analysis of Sense Tree Parsing

The sense tree parser was tested on more than 500 dictionary entries of medium and large sizes. The success rate was 91.18%, being computed as a perfect match between the output of the program and the gold standard. Furthermore, it is worth noting that an article with only one incorrect parse (i.e. one node in the sense tree attached incorrectly) was considered to be erroneously parsed altogether, an approach which disregards all the other correctly attached nodes in that entry. It is worth to mention some sources of errors and ambiguities found in the DSSD parsing for the eDTLR sense tree computing. We grouped the error sources in two main classes:

I. Inconsistencies in writing the original DTLR article

A first source of parsing errors is the non-monotony of the marker values on the same level of sense marker hierarchy:

Ex.13. **A.** [**B.** Missing] … **C.** Etc.;

Ex.14. **2.** [instead of **1.**]... **2.** Etc.;

Ex.15. **a)**… **b)** … **c)** … **b)** [instead of **d)**]etc.

The tree structure returned by the parser does not consider the consistency of each marker level. Thus, in Ex. 13, it will place the two markers **A.** And **C.** As brother nodes in the sense tree. A (partial but feasible) solution for the parser is to check the *strict monotony* of the marker neighbours, an operation which is useful also when sense markers interfere with literal enumeration.

II. Ambiguity in deciding which is the regent and which is the dependent (sub)sense

An inherent ambiguity was found for the following sequences of DTLR sense markers:

Ex.16. **1. A) b) c)** ◊ [◊]

The problem occurs since one cannot discern between attaching the first (and / or second) "◊" as depending on **c)** or on the upper level marker (**1.**). Solving these ambiguities is a problem related to the semantic contexts of the involved pairs of markers. The rule we used was to attach a "◊" to the parent level of "c)", if "c)" is the last marker in the literal enumeration, and to "c)" if the literal enumeration continues with "d)". We want to make it clear that this solution is a convention, and that it is possible that we may choose the incorrect parent for the sense represented by the "◊" marker.

Analysis of Modelling Definition Types

The segmentation of the low level sense elements, found between two successive high level markers was performed; we have found that regular expressions suffice for this particular task. The establishing of dependencies between the low level sense elements simply requires a hierarchy between the different classes of components, as proven in the solution for the task of sense tree building, and the only real difficulty is to correctly chunk the various elements. Evaluation for this task was performed using two metrics: exact match and overlap. The exact match metric represents the amount of correctly extracted chunks (in terms of *precision, recall* and *f-measure*); the overlap metric is a relaxation of the exact match metric, and represents the percentage of words assigned to the correct chunk (also in terms of *precision, recall* and *f-measure*). Since there are cases where chunks of the same type are consecutive, the first and last words of each chunk are especially marked, so as to penalize classifying all of the words as one big chunk rather than a succession of smaller chunks.

Evaluation Type	**Precision**	**Recall**	**F-measure**
Exact match	93.24%	85.41%	89.15%

Overlap	97.86%	97.80%	97.83%

Table 25: Evaluation results for low level sense element chunking

Parse Result	**Gold Standard**	**Error Rate**
Sigle	Sigle begin	29.45%
Sigle	Sigle end	28.08%
RegDef	SpecDef	6.39%
RegDef	RegDef end	3.88%
RegDef	RegDef begin	2.96%
DefExem	ItalMarker	2.73%
RegDef	SpecDef begin	2.51%
Sigle	RegDef	2.28%
RegDef	Sigle	2.05%
RegDef	SpSpecDef	2.05%

Table 26: Top ten error sources for low level sense chunking

For purposes of evaluation, we have used 52 dictionary entries of various sizes as a gold standard, totalling a number of approximately 2000 chunks and 22,000 words. The results given in Tables 25 and 26 are taken from the technical report of the eDTLR grant.

Upon further analysis of the evaluation results we have found that the most frequent errors were due to faulty *sigle* segmentation. Table 17 details the ten most frequent error types. Correcting the acquisition of *sigles* leads to a 94.43% *f-measure* for exact match and a 98.01% *f-measure* for overlap. An example of a parsed dictionary entry is given in Annex 1.

The parsing method described above has been successfully applied not only to the DLR, but also to a series of thesauri from various languages (the Romanian DAR – Dictionarul Academiei Romane, the French thesaurus – TLFI and two German thesauri – Grimm Deutsche Wortebuch and Goethe Deutsche Worterbuch) (Curteanu et al, 2010a), proving its flexibility and robustness.

The parsing method described above was applied to all of the current DLR volumes (apart from the volume containing the words starting with the letter M). Note that the DLR is the new format of the Romanian Thesaurus and is a continuation of the work started at the begging of the 20th century, and a large portion of the Romanian vocabulary has already been covered by the old format of the thesaurus, the DA (Dicţionarul Academiei), which explains why not all of the words of the Romanian language are parsed. The series words covered by the current version of the eDTLR are *D – Doznic, E – Ezredeș, K – Luzula, N – Zvugni*. This amounts to approximately 90.000 words with over 190.000 senses and more than 440.000 quotes, by far the largest structured lexical semantic database available for the Romanian language. The lexical-semantic information encoded in the electronic version of the Romanian Thesaurus can be used to automatically expand current resources such as the Romanian WordNet or the Explanatory Dictionary of Romanian (DEX), as will be shown below.

5.3. Extending the Romanian WordNet with eDTLR

Based on the electronic format of the DTLR described above, (Păpuşoi, 2009) proposes a series of heuristics that aim at automatically extending the Romanian WordNet. These heuristics make use of the extensive sense descriptions and fine sense separation of the eDTLR in order to extend existing synsets or in creating new synsets and then integrating them into the WordNet ontology. The algorithm for extending WordNet is given below.

```
For all synsets s in ROWN
1. If w in eDTLR in s then
    2. For all senses of W in ROWN
    3. If definition(w) equals definition(s) then append synset
    of s with alignment to sense of E
    4. Else, if no sense of w has a definition that matches the
    definition of s, then
        5. Extract synonyms from the definition of the current
        sense of w
        6. If any synonyms of w are found in ROWN synsets, then go
        to 2 for each of them
```

```
    7. Else create new synset for w and integrate the new
    sysnset in the ROWN taxonomy
8. Else, create new synsets for all the senses of w and
integrate them into the ROWN taxonomy
```

Figure 6: Algorithm for aligning ROWN with eDTLR

The first step in the algorithm is to attempt to align ROWN synsets to dictionary entries (step 1). If any match is found, then the algorithm attempts to match the exact eDTLR senses to ROWN synset in question (step 3). This is done on the basis of definition (gloss) matching, which is carried out using a lemma matching heuristic. The gloss of the WN sysnset and the gloss of the eDTLR sense are both lemmatized and cleaned of stop words; it is considered that the definitions match if the lemma overlap distance between them is above an empirically determined threshold. If there is no suitable candidate for gloss matching (step 4), then the algorithm attempts to match known synonyms of the current sense to literals in ROWN, by repeating steps 2 through 4 described above (steps 5 and 6). Synonyms are extracted from the gloss of the sense, as it is common for the *RegDef* portion of a DLR entry to have a segment of similar senses of other words (this procedure is called definition by synonymy by DLR authors). In case none of the synonyms are found in ROWN, a new synset is created for the current sense (step 7); the literals of the sysnset are the synonyms extracted from the gloss (if any exist), and its definition is the gloss of the sense.

The last step of the algorithm (step 8) is concerned with integrating any new synsets in the existing ROWN taxonomy. This problem is twofold: first, newly created synsets need to be compared to existing synsets, in order to avoid duplication; second, semantic relations regarding new sysnsets need to be determined. The first task is carried out using the definition matching procedure used at step 3 in the alignment algorithm described above; the threshold for definition matching is also determined empirically.

Semantic relations are determined using a set of heuristics based on key words and phrases used in eDTLR glosses. Definitional patterns for a word sense *X* such as "*Y, care...*", usually denote *hypernymy* between *X* and *Y* (for example, a person is defined as "*Fiinţă superioară, 121ocial, care se caracterizează prin ...*" – "superior, social being, which is characterized by ..." from which we can determine that the hypernym of *person* is *being*). The *holo_portion* relation can be identified by patterns of the type "*fiecare dintre cei doi X*", "*fiecare din X*", "*fiecare dintre X*" between the head word and X (for example "*Fiecare dintre pantele unui deal*" – "*each of the slopes of a hill*"), and the *holo_member* relation can be identified by patterns like "*numele mai multor X*", "*nume dat unor X*", "*nume dat unei X*", "*din familia X*" (for example "*Nume dat unor animale nevertebrate cu corpul moale*" – "*name given to a soft bodied invertebrate*").

Performance analysis has been carried out over approximately 1300 senses extracted from eDTLR. These senses were sent as input to the algorithm, and the results analyzed manually. Out of the 396 eDTLR senses aligned with existing WN synsets, 393 were correct; the output also contained 918 synsets, of which 832 were correct (90.63% accuracy). In terms of relations between sysnsets, the output contained 138 semantic links, of which 90 were correct. This shows that eDTLR can be successfully aligned to ROWN, and new synsets can be automatically added with significant accuracy.

5.4. Aligning DEX and eDTLR

Another method for aligning eDTLR and the Romanian WordNet is to use the electronic version of the Explanatory Dictionary of Romanian (DEX). The reason behind this approach is that DEX was one of the resources used to create ROWN, and the senses of DEX are aligned to WN synsets. Also, since the eDTLR senses are much more refined then those of DLR, the ROWN taxonomy can be enhanced by adding those subsenses that are missing. Because of the hierarchical structure of eDTLR entries, semantic relations between existing and new synsets can be extracted (particularly hypernym relations).

(Ivănescu, 2011) proposes a method for automatically aligning DEX and eDTLR at the sense level. The alignment procedure starts by pairing the entries of DEX to those of eDTLR by means of string matching. The result of this process is a list of paired sets, corresponding to words in both dictionaries. These words, however, may have several parts of speech (in the sense that the same word can be both a noun and a verb at the same time, for example), so the next step is to match the entries according to their part of speech. In the process of entry matching it is important to take into account the numbers attached to each head word (for example VIOÁRĂ^3), which denotes variants of polysemantic words.

Once the entries are aligned at the part of speech level, they need to be aligned at the sense level. This is done by applying three lexical distance metrics on the lemmatized sense definitions to determine sense similarity. The first of these measures is word overlap between lemmas of sense definitions, taking into account all tokens, including stop words and punctuation. The second measure is word overlap over sense definitions, disregarding stop words and punctuation. The final measure is called prefix match and it is applied over prefixes of the words in the sense definitions, disregarding stop words. The word prefixes are similar to word stems.

Based on empirical data, the result of each of the metrics is weighed down and the results are then combined, obtaining a global alignment score. If the global score is above an empirically determined threshold, then the two senses are aligned.

5.5. Conclusions

In this chapter we have analyzed methods for expanding existing lexical semantic resources for Romanian, particularly the Romanian WordNet. These resources are useful in themselves but in the context of this thesis, they are used to complement existing textual entailment resources. WordNet, in particular, is aligned to a series of important lexical-semantic databases, such as SUMO-MILO and VerbNet. Therefore, given a robust alignment between the English WordNet and its Romanian counterpart, versions of such resources as VerbNet can be ported to the Romanian language (the basic idea for creating a Romanian VerbNet has been described in chapter 3).

Another advantage of the electronic version of the Romanian Thesaurus is that it can be used to create resources for the Romanian language. At the moment, the possibility of creating a Romanian FrameNet, based on the examples available in eDTLR, is under investigation.

6. Conclusions

This thesis has described a novel way of solving textual entailment on the basis of semantic knowledge extracted from predicates and features extracted for predicate arguments. The novelty of our approach comes from the fact that we give a fresh perspective to the definition of textual entailment, which focuses on the basic elements of human utterances, predicates.

The main contributions of the thesis are given below:

- A new interpretation for the definition fo textual entailment, which is based on predicational semantics (the matching of predicates and their arguments from the text to the hypothesis) and intends to advance the field of textual entailment towards deep semantic understanding of natural language texts (first given in section 1.1, and then analyzed in chapter 3)
- A thorough description of the current state of the art in textual entailment, organized according to the type of approach towards solving TE, together with various uses of textual entailment (chapter 2)
- An original method for solving textual entailment, based on predicational semantics and argument structure unification that has been formalized within an algorithm for solving TE that relies on VerbNet and lexical taxonomies (section 3.2). The algorithm described is language independent, in the sense that, given apropriate lexical-semantic resources for a language, it can be applied directly
- A feasibility study carried out over 200 entailment pairs from the RTE-5 test set, which proves the validity of the method proposed in section 3.2 with detailed analysis of entailment pairs (section 3.3)
- A rule based system for solving textual entailment, which is an extension of a previous system on the basis of the method proposed in chapter 3 and the results obtained by this system for the RTE-5 and RTE-6 challenges, together with a detailed error analysis (chapter 4)
- A novel method for performing dictionary entry parsing, which was developed for the creation of the electronic format of the Romanian Thesaurus Dictionary (eDTLR), which can be used to enhance existing lexical-syntactic resources for

Romanian and to create new ones, allowing for the adaptation of the algorithm described in chapter 3 for the Romanian language (chapter 5)

In order to place our approach within the field of current textual entailment, we have first described in detail the current state of the domain. The description of the field of textual entailment was done in two steps: first, we have described the general notions regarding TE, with examples and definitions, together with a detailed description of the Recognizing Textual Entailment Challenge. The RTE challenge constitutes the main proving ground for current textual entailment engines, and has been so ever since its first edition, which took place in 2005; all the RTE challenges to date have been described, together with the top results and main advancements. The second step in describing the current state of the field of textual entailment is the state of the art for textual entailment engines and methods, given in chapter 2, which reviewed many of the important ideas and solutions currently available for recognizing textual entailment.

Given a thorough description of the domain, we can confidently say that the method for solving textual entailment that we put forward in chapter 3 of this thesis is both novel and effective (Moruz, 2010). It is novel because it approaches solving textual entailment in a manner that, to our knowledge, has never been used before, and it is effective because it performs well for detecting textual entailment, as we have shown in chapter 4. We have also put at the basis of our model the following interpretation of the original definition for textual entailment:

Given a text T and a hypothesis H, we say that T entails H, denoted by T→H, if and only if:

- Predicate matching: each of the predicates in H is entailed by at least one predicate in T. We say that a predicate $p \in T$ entails a predicate $q \in H$, denoted $p \rightarrow q$ if *q* is a consequence of *p*, or *p* and *q* are synonyms, or *q* is a subevent of *p*;
- Argument matching: given alignments in T for all of the predicates in H (we consider a predicate *p* in T aligned to a predicate *q* in H if there is an entailment relation between *p* and *q*), entailment holds if and only if the arguments of the aligned predicates are in an entailment relation. Two arguments are in an entailment reation if they both refer to similar entities (e.g., the heads of the arguments are synonyms, or the arguments are in a *part-of* relation, etc.), and if

the unification of their feature sets is successful and is equal to the feature set of that argument in the text. Formally, if $p \in T$ and $q \in H$ and $p \rightarrow q$ and $\arg(p) = \langle a, b \rangle$, $\arg(q) = \langle a', b' \rangle$, then entailment holds if and only if $a \rightarrow a'$ and $b \rightarrow b'$. Argument entailment is defined as the result of the unification between the argument feature structures in H and T (as described in chapter 3 in greater detail).

The implementation of this idea uses VerbNet for predicate matching and the WordNet taxonomy for extracting the argument features; we have chosen a predicate based approach because of our previous experience with predicate and verbal group structures (Moruz et al., 2006), (Curteanu et al. 2006a, 2006b, 2006c, 2007).

Chapter 4 describes the participation of our system at the RTE-5 and RTE-6 challenges, where it obtained very good results (Iftene and Moruz, 2010, 2011), thus proving the value of our approach to solving textual entailment. Also, based on our experience with information retrieval for question answering (Puscasu et al., 2007), (Iftene et al., 2008, 2009a, 2009b, 2010a, 2010b) we have adapted the entailment system with the addition of an information retrieval element, in order to solve the RTE-5 pilot task.

Because of the way in which the idea described above is implemented, it can be easily adapted for the Romanian language, given the basic resources needed for the system operation (WordNet and VerbNet). Based on our experience with dictionary entry parsing, which resulted in the electronic format of the Romanian Thesaurus Dictionary (DLR) (Curteanu et al. 2008a, 2008b)(Curteanu et al 2009)(Curteanu et al., 2010a, 2010b), we propose the extension of current resources for Romanian (such as WordNet) and the creation of new ones on the basis of existing alignments (VerbNet).

In terms of future work, our immediate goal is to take part in the seventh Recognizing Textual Entailment Challenge, which will take place in September 2011, with a further improved system. We also intend to make this system widely available as either a web service or an open source package.

For longer term goals, we intend to adapt and port the entailment system described in this thesis (particularly in chapter 4) for the Romanian language. As discussed above, a Romanian version of our entailment system is dependent on a series of Romanian lexical-semantic resources, such as WordNet or VerbNet, which can be automatically created or extended with reasonable accuracy by using the electronic format of the Romanian Thesaurus Dictionary. At the moment we have carried out limited studies to determine the feasibility of extending the Romanian WordNet, and are currently exploring the possibility of creating a Romanian FrameNet on the basis of examples in DLR.

In terms of applications for our entailment system, we have been exploring two possibilities: validating legal documents (checking for entailment between various laws, in order to determine possible contradictions or discrepancies) and opinion mining from Amazon costumer reviews (we formulate the problem of determining opinion as an entailment problem by creating hypotheses from common user searches: “I want a camera with detachable lens”→”H: Camera *X* has a detachable lens”).

A major use for textual entailment is the validation of machine summaries; given a summary of a text, or a collection of texts, this summary is valid if and only if it is entailed by the original source of the summary. Since it is difficult to handle entailment for large hypotheses (hypotheses with more than one sentence), we propose the splitting of the summary to its component sentences, thus reducing the problem to solving textual entailment in a summarization setting, as defined by the RTE-5 challenge pilot task and the RTE-6 challenge main task.

Textual entailment can also be used for knowledge base population, and this idea is also explored in the RTE-6 challenge pilot task. The KBP Validation Pilot aims at validating the output of the systems participating in the KBP Slot Filling task[24] by using Textual Entailment techniques.

The KBP Slot Filling task is focused on searching a collection of newswire and Web documents and extracting values for a pre-defined set of attributes (“slots”) for a series of entities. Given an entity in a knowledge base and an attribute for that entity, systems must find, in a large corpus, the correct value(s) for that attribute and return the extracted information together with a corpus document supporting it as a correct slot filler. The RTE KBP Validation Pilot is based on the assumption that an extracted slot filler is correct if and only if the supporting

[24] http://nlp.cs.qc.cuny.edu/kbp/2011/

document entails an hypothesis created on the basis of the slot filler (Textual Entailment has previously been used as a tool for validating outputs of other applications, as was the case of the Answer Validation Exercise within CLEF (Peñas et al., 2007)).

The KBP Validation Task consists of determining whether a candidate slot filler is supported in the justification document. Each slot filler that is submitted by a system participating in the KBP Slot Filling task creates one evaluation item (i.e. a T-H “pair”) for the RTE-KBP Validation Pilot, where T is the source document that was cited as supporting the slot filler, and H is a set of simple, synonymous Hypotheses created from the slot filler. The resulting T-H pairs differ from the traditional pairs in that T is an entire document (vs. single sentences or paragraphs) and H is a set of roughly synonymous sentences representing different linguistic realizations of the same slot filler (some of which may be ungrammatical)[25].

Logic based methods are hindered by the difficulty of transforming natural language text into a logical formula (in propositional logic). Fuzzy logic, however, is inherently flexible, and could thus be used with greater success for modelling natural language phenomena. Because of this, we intend to examine the eefect of using fuzzy logic modelling techniques or solving textual entailment.

One of the main goals of textual entailment is to provide machine understanding for natural language at a level close to teh human one. In order to achieve this, an entailment system needs to access the deep semantic level of utterances. We are currently examining a novel way of representing the semantic knowledge extracted from the text and the hypothesis, which would lead to deep semantic understanding of text, and is based on the idea of semantic triples represented in a graph format.

The idea of a semantic triple is to determine the semantic relation between each of the entities in an utterance, and create the semantic triple $rel(x, y)$, where *rel* is the relation between the entities x and y. If we consider the entities nodes and the relations named arcs, the semantic representation of a text becomes a connected graph, which has subgraphs clustered around separate predicates as semantic cliques. Given such a representation, entailment can be defined as an isomorphism between the graph of the hypothesis and a subgraph of the text. We are

[25] The description of the RTE-6 KBP pilot task was taken from the RTE-6 challenge guidelines (http://www.nist.gov/tac/2010/RTE/RTE6_KBP_Validation_Pilot_Guidelines.pdf)

convinced that a semantic representation of this type allows for easier detection of isomorphism and a more relevant match over semantic graphs.

The difficulty of this approach arises from the necessity of precise semantic analysis over human utterances, such as words sense disambiguation, coreference resolution or semantic role labelling. It may also become necessary to create semantic nodes that have no surface realization, which further increases the difficulty of such an approach.

7. Bibliography

Adams, R. 2006. *Textual Entailment Through Extended Lexical Overlap*. The Second PASCAL Recognising Textual Entailment Challenge. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, Pages 128-132, 10 April, 2006, Venice, Italia

Adams, R., Nicolae, G., Nicolae, C. and Harabagiu, S. 2007. *Textual Entailment Through Extended Lexical Overlap and Lexico-Semantic Matching.* . In Proceedings of the –ACL PASCAL Workshop on Textual Entailment and Paraphrasing. Pp.119-124. 28-29 June, Prague, Czech Republic

Agerri, R. 2008. *Metaphor in Textual Entailment*, In Companion Procs. of the 22nd Int. Conf. on Computational Linguistics (COLING 2008), pp.2-6, Manchester, UK.

Aharon, R. B., Szpektor, I., Dagan, I. 2010. *Generating Entailment Rules from FrameNet*. Proceedings of the ACL 2010 Conference Short Papers, pages 241–246, Uppsala, Sweden, 11-16 July 2010.

Akhmatova, E. 2005. *Textual Entailment Resolution via Atomic Propositions*. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages 61-64, 33–36 April, 2005, Southampton, U.K

Baker, C. F., Fillmore, C. J., and Lowe, J. B. 1998. *The Berkeley FrameNet project*. In Proceedings of the 17th International Conference on Computational Linguistics (COLING/ACL-98), pages 86–90, Montreal.

Bayer, S., Burger, J., Ferro, L., Henderson, J., Yeh, A. 2005. *MITRE's Submissions to the EU Pascal RTE Challenge*. In Proceedings of the First PASCAL Challenge Workshop for Recognising Textual Entailment, pages 41–44, 11–13 April, 2005, Southampton, U.K.

Bensley, J. and Hickl, A. 2008. *Workshop: Application of LCC's GROUNDHOG System for RTE-4*. In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, 2008. Gaithersburg, Maryland, USA.

Bentivogli, L., Magnini, B., Dagan, I., Dang, H.T. Giampiccolo, D. 2010. *The Fifth PASCAL Recognizing Textual Entailment Challenge*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Bentivogli, L., Clark, P., Dagan, I., Dang, H. T., Giampiccolo, D. 2011. *The Sixth PASCAL Recognizing Textual Entailment Challenge*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Berant, J., Dagan, I. and Goldberger, J. 2010. *Global Learning of Focused Entailment Graphs*. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1220–1229, Uppsala, Sweden, 11-16 July 2010.

Bergmair, R. 2008. *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.

Blake, C., Zheng, W., Painter, K., Weyerhaeuser, W. 2011. *The Role of Semantics in Recognizing Textual Entailment*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Bos, J. and Markert, K. 2005. *Recognising Textual Entailment with Logical Inference*. In HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Pages 628-635, Association for Computational Linguistics, Morristown, NJ, USA.

Breck, E. 2010. *A simple system for detecting non-entailment*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Bunke, H. and Shearer, K. 1998. *A graph distance metric based on the maximal common subgraph*. Pattern Recogn. Lett., 19(3-4), pages 255–259.

Burchardt, A. and Frank, A. 2006. *Approximating Textual Entailment with LFG and FrameNet Frames.* Proceedings of the 2nd PASCAL Workshop on Recognising Textual Entailment, 92-97, Venice, Italy.

Burchardt, A., Reiter, N., Thater, S., Frank, A. 2007. *A Semantic Approach To Textual Entailment: System Evaluation and Task Analysis*. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Pages 10-15. 28-29 June, Prague, Czech Republic.

Cabrio, E., Magnini, B. 2010. *Toward Qualitative Evaluation of Textual Entailment Systems*. COLING (Posters) 2010, pp: 99-107

Castillo, J. J. 2010a. *Recognizing Textual Entailment: Experiments with Machine Learning Algorithms and RTE Corpora*. Research in Computing Science. Special Issue: Natural Language Processing and its Applications, pp: 155-166

Castillo, J. J. 2010b. *Sagan in TAC2009: Using Support Vector Machines in Recognizing Textual Entailment and TE Search Pilot task*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Castillo, J. J. 2011c. *A machine learning approach for recognizing textual entailment in Spanish*, Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas, pp: 62-67

Castillo, J. J. 2011. *Sagan in TAC2010: A Machine Learning Approach to RTE within a Corpus*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Chambers, N., Cer, D., Grenager, T., Hall, D., Kiddon, C., MacCartney, B., Marneffe, M. C., Ramage, D., Yeh, E., Manning, C. D. 2007. *Learning Alignments and Leveraging Natural Logic*. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Pp.165-170. 28-29 June, Prague, Czech Republic

Chklovski, T. and Pantel, P. 2004. *Verbocean: Mining the web for fine-grained semantic verb relations*. In Proceedings of EMNLP 2004, pages 33–40, Barcelona, Spain, July. Association for Computational Linguistics.

Clark, P. and Harrison, P. 2008. *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, 2008. Gaithersburg, Maryland, USA.

Clark, P., Harrison, P. 2010. *An Inference-Based Approach to Recognizing Entailment*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Clark, P., Harrison, P. 2011. *BLUE-Lite: a Knowledge-Based Lexical Entailment System for RTE6*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Cristea, D., Răschip, M., Forăscu, C., Haja, G., Florescu, C., Aldea, B., Dănilă, E. 2007. *The Digital Form of the Thesaurus Dictionary of the Romanian Language*. In Proceedings of the 4th International IEEE Conference SpeD 2007, "Advances in Spoken Language Technology", C. Burileanu, H.N. Teodorescu (Eds.), pp. 195-206. Iasi, 10-12 Mai 2007, Romanian Academy Publishing House, ISBN 978-973-27-1516-1.

Cristea, D., Răschip M., **Moruz A.**,(2009) *Steps in building the Electronic Version of a Thesaurus Dictionary of the Romanian Language*, a treia conferinţă a Academiei de Ştiinţe Tehnice din România, in BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI, Universitatea Tehnică „Gheorghe Asachi” din Iaşi, Secţia *MATEMATICĂ. MECANICĂ TEORETICĂ. FIZICĂ

Curteanu, N., **Moruz, M.**, Trandabat, D., Bolea, C., Dornescu I. 2006a, *The Structure and Parsing of Romanian Verbal Group and Predicate* 4th European Conference on Intelligent Systems and Technologies, ECIT 2006, Iaşi, România, 2006, ISBN 973-730-265-6

Curteanu, N., Trandabat, D., **Moruz M.** 2006b, *Substructures of the (Romanian) Predicate and Predication using FX-bar Projection Functions on the Syntactic Interface* 4th European Conference on Intelligent Systems and Technologies, ECIT 2006, Iaşi, Romania, 2006, ISBN 973-730-265-6

Curteanu, N., Trandabat, D., **Moruz, M.** 2006c, *Structura grupului verbal, predicaţia lexicală şi reprezentarea logică a predicatului în limba română*, Forăscu, C., Tufiş, D., Cristea, D. (Eds.), Lucrările atelierului „Resurse lingvistice şi instrumente pentru prelucrarea limbii române", Iaşi, 3-4 Noiembrie 2006, Editura Universităţii "Al.I.Cuza" Iaşi, ISBN 978-973-703-208-9, pag. 143-148, Iaşi, Romania 2006

Curteanu, N., Trandabăţ, D., **Moruz, M. A.** 2007. *Functional FX bar Projections of the (Romanian) Verbal Group and Sub-Groups on the Syntactic-Semantic Interface*, Journal of Computer Science of Moldova, Academy of Science of Moldova, Institute of Mathematics and Computer Science, volume 15, no. 2(44)/2007, pp 123 - 152.

Curteanu, N., Trandabăţ, D., **Moruz, A.**, Pavel, G., Vereştiuc, C. 2008a. *Dictionary Sense Segmentation & Dependency – A Flexible Strategy for Thesauri Parsing*, Proceedings of 5th European Conference on Intelligent Systems and Technologies ECIT 2008, CD edition.

Curteanu, N., **Moruz, A.**, Trandabăţ, D., 2008b. *Extracting Sense Trees from the Romanian Thesaurus by Sense Segmentation & Dependency Parsing*, Proceedings of CogAlex Cognitive Aspects of the Lexicon: Enhancing the Structure, Indexes and Entry Points of Electronic Dictionaries, COLING 2008 Workshop endorsed by SIGLEX, pp. 55–63, ISBN 978-1-905593-56-9

Curteanu, N., **Moruz, A.**, Trandabăţ, D., Bolea, C., Spătaru, M., Husarciuc, M. 2009. *Sense tree parsing and definition segmentation in eDTLR Thesaurus*, (in Romanian), Trandabăţ, D., Tufis, D., Cristea, D. (Eds.), Proceedings of the Workshop "Linguistic Resources and Instruments for Romanian Language Processing", pp. 65-74. Iasi, Romania, November, 19-21, 2008, "Al.I.Cuza" University Publishing House, ISSN 1843-911X.

Curteanu, N., **Moruz, A.**, Trandabăţ D. 2010a. *An Optimal and Portable Parsing Method for Romanian, French, and German Large Dictionaries* Proceedings of the 2nd Workshop on Cognitive Aspects of the Lexicon (CogALex 2010), pages 38–47, Beijing, August 2010

Curteanu, N., Trandabăţ D., **Moruz, M. A**. 2010b. *Lexical semantics modeling and robust parsing for Romanian, French and German Thesauri* (in Romanian), Adrian Iftene, Dan

Cristea and Dan Tufiș (Eds.), Proceedings of ConsILR 2010, Univ. Al. I. Cuza Press, pp. 113-122. Bucharest, Romania, May 2010, ISSN 1843-911X.

Dagan, I. and Glickman, O. 2004. *Probabilistic textual entailment: Generic applied modeling of language variability*. In Learning Methods for Text Understanding and Mining, Grenoble, France.

Dagan, I., Magnini, B., Glickman, O. 2005. *The PASCAL Recognising Textual Entailment Challenge*. In Proceedings of the First PASCAL Challenge Workshop for Recognising Textual Entailment, pages 1–9, 11–13 April, 2005, Southampton, U.K.

Dinu, G., Wang, R. 2009. *Inference rules and their application to recognizing textual entailment*, Proceedings of the 12th Conference of the European Chapter of the ACL, pages 211–219, Athens, Greece, 30 March – 3 April 2009.

Delmonte, R., Tonelli, S., Aldo Piccolino Boniforti, M., Bristot, A., Pianta, E. 2005. *VENSES – a Linguistically-Based System for Semantic Evaluation*. In Proceedings of the First PASCAL Challenge Workshop for Recognising Textual Entailment, pages 49–52, 11–13 April, 2005, 129 Southampton, U.K.

Delmonte, R., Tonelli, S., Tripodi, R. 2010. *Semantic Processing for Text Entailment with VENSES*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Erjavec, T, Evans, R., Ide, N., Kilgariff A., (2000): The CONCEDE Model for Lexical Databases. Research Report on TEI-CONCEDE LDB Project, Univ. of Ljubljana, Slovenia.

Fellbaum, C., editor. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech and Communications. MIT Press, Cambridge, Massachusetts.

Ferrández, O., Muñoz, R., Palomar. M. 2010. *Alicante University at TAC 2009: Experiments in RTE*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Ferrés, D., Rodríguez, H. 2007. *Machine Learning with Semantic-Based Distances Between Sentences for Textual Entailment*. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Pp.60-65. 28-29 June, Prague, Czech Republic

Glickman, O., Dagan, I. and Koppel, M. 2005. *Web Based Probabilistic Textual Entailment*. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages 17-20, 33–36 April, 2005, Southampton, U.K

Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B. 2007. *The Third PASCAL Recognising Textual Entailment Challenge*. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Pp.1-9. 28-29 June, Prague, Czech Republic

Giampiccolo, D., Dang, H. T., Magnini, B., Dagan, I., Dolan, B. 2008. *The Fourth PASCAL Recognising Textual Entailment Challenge*. In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, 2008. Gaithersburg, Maryland, USA.

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B. and Szpektor, I. 2006. *The Second PASCAL Recognising Textual Entailment Challenge*, In Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy.

Bar-Haim, R., Berant, J., Dagan, I., Greental, I., Mirkin, S., Shnarch, E., Szpektor, I. 2008. *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, 2008. Gaithersburg, Maryland, USA.

Bar-Haim, R., Berant J., and Dagan, I. 2009. *A compact forest for scalable inference over entailment and paraphrase rules*. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1056–1065, Singapore, 6-7 August 2009.

Harabagiu, S., Pasca, M., and Maiorano, S. 2000. *Experiments with Open-Domain Textual Question Answering*. Proceedings of the 18th conference on Computational linguistics, pp: 292-298, Beijing, China.

Harmeling, S. 2007. *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. Pages 137-142. 28-29 June, Prague, Czech Republic.

Heilman, M. and Smith, N. A. 2010. *Tree Edit Models for Recognizing Textual Entailments, Paraphrases, and Answers to Questions*. The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp 1011-1019.

Herrera, J., Peas, A. and Verdejo, F. 2005. *Textual Entailment Recognition Based on Dependency Analysis and WordNet*. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages 21-24, 33–36 April, 2005, Southampton, U.K

Hickl, A., Bensley, J., Williams, J., Roberts, K., Rink, B., Shi, Y. 2006. *Recognising Textual Entailment with LCC's GROUNDHOG System*. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, Pages 80-85, 10 April, 2006, Venice, Italia

Hickl, A. and Bensley, J. 2007. *A Discourse Commitment-Based Framework for Recognising Textual Entailment*. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Pages 185-190. 28-29 June, Prague, Czech Republic.

Hickl, A. 2008. *Using Discourse Commitments to Recognize Textual Entailment*. Proceedings of the 22nd Conference on Computational Linguistics (COLING 2008). Pp: 337-344. Manchester, United Kingdom.

Iftene, A., Balahur-Dobrescu, A. 2007. *Hypothesis Transformation and Semantic Variability Rules Used in Recognising Textual Entailment*. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Pages 125-130. 28-29 June, Prague, Czech Republic.

Iftene, A., Balahur-Dobrescu, A. 2008. *Named Entity Relation Mining Using Wikipedia*. In Proceedings of the Sixth International Language Resources and Evaluation (LREC'08). pp: 763-766. 28-30 May, Marrakech, Morocco.

Iftene, A. 2008. UAIC Participation 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.

Iftene, A., Trandabăţ, D., Pistol, I., **Moruz, A.**, Balahur-Dobrescu, A., Cotelea, D., Dornescu, I., Drăghici, I., Cristea, D. 2008. *UAIC Romanian Question Answering system for QA@CLEF*, Lecture Notes in Computer Science, 5152/2008, p. 336-34.

Iftene, A., Trandabăţ, D., Pistol, I., **Moruz, A.**, Husarciuc, M., Cristea, D. 2009a. *UAIC Participation at QA@CLEF2008*, Evaluating Systems for Multilingual and Multimodal Information Access, Lecture Notes in Computer Science, vol. 5706/2009, pp. 385-392, ISBN 978-3-540-74998-1, ISSN 0302-9743 (Print) 1611-3349

Iftene, A., Trandabăţ, D., Pistol, I., **Moruz, A.M.**, Husarciuc, M., Sterpu, M. and Turliuc, C. 2009b. *Question Answering on English and Romanian Languages*, Working Notes for the CLEF 2009 Workshop, pp: 229-236. Corfu, Greece, 30 Sept. - 2 Oct. 2009.

Iftene, A. 2009. *Textual Entailment*, PhD Thesis, "Al. I. Cuza" University, Iasi

Iftene A., Trandabăţ D., Husarciuc M., **Moruz A.** 2010a. *Question Answering on Romanian, English and French Languages*, in CLEF 2010 LABs and Workshops, Notebook Papers, 22-23 September 2010, Padua, Italy, ISBN 978-88-904810-0-0.

Iftene A., Trandabăţ D., **Moruz A.**, Pistol I., Husarciuc M., Cristea D. 2010b. *Question Answering on English and Romanian Languages*, in Multilingual Information Access Evaluation Vol. I Text Retrieval Experiments, LNCS Series, Eds. Carol Peters, Giorgio Di Nunzio, Mikko Kurimo, Thomas Mandl, Djamel Mostefa, Anselmo Penas, Giovanna Roda, Springer, 2010, pp. 229-236.

Iftene, A., **Moruz, M.-A**. 2010. *UAIC Participation at RTE5*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Iftene, A., **Moruz, M.-A**. 2011. *UAIC Participation at RTE6*, *The Sixth PASCAL Recognizing Textual Entailment Challenge*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Inkpen, D., Kipp, D. and Nastase, V. 2006. *Machine Learning Experiments for Textual Entailment*. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, Pages 17-20, 10 April, 2006, Venice, Italia

Ivănescu, M. L. 2011. *Tehnici de completare a wordnetului românesc din Dicţionarul Tezaur al limbii române*. B. Sc. Thesis, Faculty of Computer Science, “Al. I. Cuza” University, Iaşi, Romania

Jia, H., Huang, X., Ma, T., Wan, X., Xiao, J. 2011. *PKUTM Participation at TAC 2010 RTE and Summarization Track*, *The Sixth PASCAL Recognizing Textual Entailment Challenge*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Jijkoun, V. and de Rijke, M. 2005. *Recognising Textual Entailment Using Lexical Similarity*. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, 25–28 April, 2005, Southampton, U.K.

Joachims, T. 2002. *Learning to Classify Text Using Support Vector Machines*. Kluwer.

Jurafsky, D. and Martin, J. H. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing*, Computational Linguistics and Speech Recognition. Prentice-Hall. Chapter 14-15.

Kammerer, M. (2000): *Wöterbuchparsing Grundsätzliche Überlegungen und ein Kurzbericht über praktische Erfahrungen*, http://www.matthias-kammerer.de/content/WBParsing.pdf

Kamp, H. and Reyle, U. 1993. *From Discourse to Logic*. Introduction to Model theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Kluwer, Dordrecht, Netherlands.

Katrenko, S. and Adriaans, P. 2006. *Using Maximal Embedded Syntactic Subtrees for Textual Entailment Recognition*. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, Pages 33-37, 10 April, 2006, Venice, Italia

Kipper, K., Trang Dang, H., and Palmer, M. 2000. *Class-based construction of a verb lexicon*. In AAAI/IAAI, pages 691–696.

Kipper-Schuler, K. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, Computer and Information Science Dept., University of Pennsylvania, Philadelphia, PA, June

Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. 2006. *Extending VerbNet with Novel Verb Classes*. Fifth International Conference on Language Resources and Evaluation (LREC 2006). Genoa, Italy. June, 2006.

Korhonen, A. and Briscoe, T. 2004. *Extended Lexical-Semantic Classification of English Verbs*. In Proceedings of the HLT/NAACL Workshop on Computational Lexical Semantics, pp: 38-45, Boston, MA

Kouylekov, M. and Magnini., B. 2005. *Recognising Textual Entailment with Tree Edit Distance Algorithms*. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages 17-20, 25–28 April, 2005, Southampton, U.K.

Kouylekov, M., Negri, M., Mehdad, Y., Cabrio, E. 2011. *FBK Participation in RTE6: Main and KBP Validation Task*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Kozareva, Z. and Montoyo, A. 2006. *MLEnt: The Machine Learning Entailment System of the University of Alicante*. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, Pages 17-20, 10 April, 2006, Venice, Italia

Krestel, R., Witte, R., Bergle, S. 2010. *Believe It or Not: Solving the TAC 2009 Textual Entailment Tasks through an Artificial Believer System*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Lemnitzer, L., Kunze, C. 2005: *Dictionary Entry Parsing*, ESSLLI 2005

Levin, B. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*, University of Chicago Press, Chicago, IL.

Li, B., Irwin, J., Garcia, E.V. and Ram, A. 2007. *Machine Learning Based Semantic Inference: Experiments and Observations at RTE-3*. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Pages 159-164. 28-29 June, Prague, Czech Republic.

Li, F., Zheng, Z., Tang, Y., Bu, F., Ge, R., Zhu, X., Zhang, X., Huang, M. 2008. *THU QUANTA at TAC 2008 QA and RTE track*. In Text Analysis Conference (TAC 2008) Workshop -

RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, 2008. Gaithersburg, Maryland, USA.

Litkowski, K. 2010. *Overlap Analysis in Textual Entailment Recognition*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

MacCartney, B., Grenager, T., de Marneffe, M. C., Cer, D. and Manning, C. D. 2006. *Learning to recognize features of valid textual entailments*. In Proceedings of NAACL, pages 41–48, New York City, NY.

MacKinlay, A., Baldwin, T. 2010, *A Baseline Approach to the RTE5 Search Pilot*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Majumdar, D., Bhattacharyya, P.. 2011. *Lexical Based Text Entailment System for Main Task of RTE6*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Malakasiotis, P. 2010. *AUEB at TAC 2009*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Mehdad, Y., Cabrio, E., Negri, M., Kouylekov, M., Magnini, B. 2010a. *Using Lexical Resources in a Distance-Based Approach to RTE,* Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Mehdad, Y., Moschitti, A., Zanzotto, F. M. 2010b. *SemKer: Syntactic/Semantic Kernels for Recognizing Textual Entailment*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Mehdad, Y.; Moschitti, A. & Zanzotto, F. M. 2010c. *Syntactic/Semantic Structures for Textual Entailment Recognition*. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 1020–1028, Los Angeles, California, June 2010.

Mehdad, Y., Negri, M., Frederico, M. 2010d. *Towards cross-lingual textual entailment*, Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 321–324, Los Angeles, California, June 2010.

Miller, G. 1995. *WordNet: A Lexical Database for English*, Communications of the ACM, 38(11): Pages 39-41.

Mirkin, S., Specia, L., Cancedda, N., Dagan, I., Dymetman, M. and Szpektor, I. 2009. *Source-Language Entailment Modeling for Translating Unknown Terms*. Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 791–799, Suntec, Singapore, 2-7 August 2009.

Mirkin, S., Bar-Haim, R., Dagan, I., Shnarch, E., Stern, A., Szpektor, I., Berant, J. 2010. *Addressing Discourse and Document Structure in the RTE Search Task*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Mirkin, S., Dagan, I., Pado, S. 2010a. *Assessing the Role of Discourse References in Entailment Inference*, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1209-1219

Mirkin, S., Berant, J., Dagan, I. and Shnarch, E. 2010b. *Recognising Entailment within Discourse*. Proceedings of the 23rd International Conference on Computational Linguistics, pp: 770-778, Beijing, China.

Moens, M. and Steedman, M. 1988. *Temporal Ontology and Temporal Reference*. Computational Linguistics, 14:15–38.

Montejo-Ráez, A., Perea, J.M, Martínez-Santiago, F., García-Cumbreras, M. Á., Martín-Valdivia, M., Ureña-López, A. 2007. *Combining Lexical-Syntactic Information with Machine Learning for Recognising Textual Entailment*. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Pages 78-82. 28-29 June, Prague, Czech Republic.

Montejo-Ráez, A., Perea-Ortega, J. M., Martínez-Santiago, F., Ureña-López, L. A. 2011. *SINAI at RTE-6: Personalized Page Rank Vectors and Named Entities*, Proceedings of the Third

Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Moruz, M., Curteanu, N., Trandabat, D., Dornescu, I., Bolea C., 2006. *Parsarea predicatului (verbal / nominal) şi a clauzei (finite / nefinite) în limba română. Aplicare la parsarea FDG.* Forăscu, C., Tufiş, D., Cristea, D. (Eds.), Lucrările atelierului „Resurse lingvistice şi instrumente pentru prelucrarea limbii române", Iaşi, 3-4 Noiembrie 2006, Editura Universităţii "Al.I.Cuza" Iaşi, ISBN 978-973-703-208-9, pag. 123-128, Iaşi, Romania 2006

Moruz, M. A. 2010. *Folosirea verbelor pentru determinarea inferenţelor textuale*, Lucrările conferinţei Resurse lingvistice şi instrumente pentru prelucrarea limbii române, Iftene A., Teodorescu, H.N., Tufiş, D. eds., pp. 223-233.Bucureşti, 6-7 mai 2010.

Neff, M., Boguraev, B. (1989) *Dictionaries, Dictionary Grammars and Dictionary Entry Parsing*, Proc. of the 27th annual meeting on Association for Computational Linguistics pp: 91 – 101, Vancouver, British Columbia, Canada

Niles, I. and Pease, A. 2001. *Towards a standard upper ontology*. In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems FOIS-2001, pp:2-9.

Ofoghi, B., Yearwood, J. 2010. *UB.dmirg: A Syntactic Lexical System for Recognizing Textual Entailments*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Ofoghi, B., Yearwood, J. 2011, UB.dmirg: *Learning Textual Entailment Relationships Using Lexical Semantic Features*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Padó, S., de Marneffe, M, MacCartney, B., Rafferty, A., Yeh, E., Manning, C. 2008. *Deciding Entailment and Contradiction with Stochastic and Edit Distance-based Alignment*, In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of

Standards and Technology (NIST). November 17-19, 2008. Gaithersburg, Maryland, USA.

Padó, S., Galley, M., Jurafsky , D., Manning, C. 2009. *Robust Machine Translation Evaluation with Entailment Features*, Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp: 297-305

Palmer, M., Gildea, D., Kingsbury, P. 2005. *The Proposition Bank: An Annotated Corpus of Semantic Roles*, Computational Linguistics, 31:1 , pp. 71-105.

Pakray, P., Bandyopadhyay, S., Gelbukh, A. 2010. *Lexical based two-way RTE System at RTE-5*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Pakray, P., Pal, S., Poria, S., Bandyopadhyay, S., Gelbukh, A. 2011. *JU_CSE_TAC: Textual Entailment Recognition System at TAC RTE-6*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Papineni, K., Roukos, S., Ward, T. and Zhu, W. 2001. *BLEU: a method for automatic evaluation of machine translation*. Research report, IBM.

Pazienza, M. T., Pennacchiotti, M., Zanzotto, F M. 2005. *Textual Entailment as Syntactic Graph Distance: a rule based and a SVM based approach*. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages 9-12, 25–28 April, 2005, Southampton, U.K.

Păpuşoi, Ş. I. 2009. *Tehnici automate de completare a wordnetului românesc din Dicţionarul Tezaur al Limbii Române*, B. Sc. Thesis, Faculty of Computer Science, “Al. I. Cuza” University, Iaşi, Romania

Peñas, A., Rodrigo, Á., Verdejo, F. 2007. *Overview of the Answer Validation Exercise 2007*. In Working Notes of the CLEF 2007 Workshop, pp: 237-248. 19-21 September, Budapest, Hungary.

Pérez, D. and Alfonseca, E. 2005. *Application of the Bleu algorithm for recognising textual entailments.* In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages 9-12, 11–13 April, 2005, Southampton, U.K.

Perini, A. 2011. *Detecting Textual Entailment with Conditions on Directional Text Relatedness Scores*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Perini, A. 2011. *Detecting Textual Entailment with Conditions on Directional Text Relatedness Scores Revisited*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Puscasu, G., Iftene, A., Pistol, I., Trandabăţ, D., Tufiş, D., Ceauşu, A., Stefănescu, D., Ion, R., Dornescu, I., **Moruz, A.**, Cristea, D. 2007. *Cross-Lingual Romanian to English Question Answering at CLEF 2006*. Evaluation of Multilingual and Multi-modal Information Retrieval, 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006, Alicante, Spain, September 20-22, 2006, Revised Selected Papers, Lecture Notes in Computer Science vol. 4730/2007, pag. 385-394, ISBN 978-3-540-74998-1, ISSN 0302-9743 (Print) 1611-3349 (Online)

Quinlan, J.R. 2000. *C5.0 Machine Learning Algorithm* http://www.rulequest.com

Raina, R., Haghighi, A., Cox, C., Finkel, J., Michels, J., Toutanova, K., MacCartney, B., Marneffe, M.C., Manning, C., Ng, A., Y. 2005. *Robust Textual Inference using Diverse Knowledge Sources*. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages 57-60, 11–13 April, 2005, Southampton, U.K.

Ren, H., Ji, D., Wan, J. 2010. WHU at TAC 2009: *A Tri-categorization Approach to Textual Entailment Recognition*, Proceedings of the Second Text Analysis Conference (TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Rodriguez, H., Climent, S., Vossen, P., Blomska, L., Peters, W., Alonge, A., Bertagna, F., Roventini, A. 1998. The *Top-Down Strategy for Building EuroWordNet: Vocabulary*

Coverage, Base Concepts and Top Ontology, Computers and the Humanities, 32(2-3), pp: 117-152.

Sacaleanu, B., Orasan, C., Spurk, C., Ou, S., Ferrandez, O., Kouylekov, M. and Negri, M. 2008 *Entailment-based Question Answering for Structured Data*. In Proceedings of Coling 2008: Companion volume: Posters and Demonstrations, pp. 29 – 32. Manchester, UK, August,

Sammons, M., Vydiswaran, V. G. V. and Roth, D. 2010. *Ask not what Textual Entailment can do for you...*. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp: 1199-1208

Shivhare, H., Nath, P., Jain, A. 2011. *Semi Cognitive approach to RTE 6 - Using FrameNet for Semantic Clustering*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Stamou, S., Oflazer, K., Pala, K., Christodoulakis, D., Cristea, D., Tufis, D., Koeva, S., Totkov, G., Duitot, D., Grigoriadou, M., *BALKANET - A Multilingual Semantic Network for the Balkan Languages*, in Proceedings of the International WordNet Conference, pp: 21-25. Mysore, India, January 21-25, 2002.

Szpektor, I. and Dagan, I. 2008. *Learning entailment rules for unary templates*, Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 849–856. Manchester, August 2008

Stern, A., Shnarch, E., Mirkin, S., Kotlerman, L., Zeichner, N., Dagan, I., Lotan, A., Berant, J. 2011. *Rule Chaining and Approximate Match in textual inference*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Tateishi, K., Ishikawa, K. 2011. *IKOMA at TAC2010: Textual Entailment System using Local-Novelty Detection*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Tatu, M., Iles, B., Slavick, J., Novischi, A., Moldovan, D. 2006. *COGEX at the Second Recognising Textual Entailment Challenge*. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, Pages 17-20, 10 April, 2006, Venice, Italia

Tatu, M. and Moldovan, D. 2007. *COGEX at RTE3*. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Pages 22-27. 28-29 June, Prague, Czech Republic.

Tătar, D., Şerban, G., Mihiş, A. and Mihalcea, R. 2009. *Textual entailment as a directional relation*. Journal of Research and Practice in Information Technology, 41(1): 53-64, 2009.

Tufiş, Dan 2001: From Machine Readable Dictionaries to Lexical Databases, RACAI, Romanian Academy, Bucharest, Romania.

Tufiş, D., Barbu, E., Barbu-Mititelu, V., Ion, R., and Bozianu, L. 2004. *The Romanian Wordnet*. In Dan Tufiş (ed.), Romanian Journal on Information Science and Technology. Special Issue on BalkaNet, volume 7, pp. 105-122. Romanian Academy, April 2004. ISSN 1453-8245.

Tufiş, D. 2008. *Ro-WordNet: ontologie lexicală pentru limba română*. Academica, XVIII (208-209): 30-34, februarie-martie 2008. Editura Academiei Române. ISSN 1220-5737.

Tufiş, D., Ştefãnescu, D. 2011. *Connotavive Meaning Markup for WordNet*. Proceedings of the Romanian Academy, Series A, Volume 12, Number 2/2011, pp. 143–150

Volokh, A., Neumann, G., Sacaleanu, B. 2011. *Combining Deterministic Dependency Parsing and Linear Classification for Robust RTE*, Proceedings of the Third Text Analysis Conference (TAC 2010) November 15-16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Wang, R. and Neumann, G. 2008. An Accuracy-Oriented Divide-and-Conquer Strategy for Recognising Textual Entailment. In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, 2008. Gaithersburg, Maryland, USA.

Wang, R., Zhang, Y., Neumann, G. 2010. *A Joint Syntactic-Semantic Representation for Recognizing Textual Relatedness*, Proceedings of the Second Text Analysis Conference

(TAC 2009) November 16-17, 2009 National Institute of Standards and Technology Gaithersburg, Maryland, USA

Wang, R., Callison-Burch, C., 2010, *Cheap Facts and Counter-Facts*, Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pp: 163-167

Wu, D. 2005. *Textual Entailment Recognition Based on Inversion Transduction Grammars*. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages 13-16, 37–40 April, 2005, Southampton, U.K.

XCES TEI Standard, Variant P5 (2007): http://www.tei-c.org/Guidelines/P5/

Zanzotto, F. M., Moschitti, A., Pennacchiotti, M., Pazienza, M. T. 2006. *Learning textual entailment from examples*. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, Pages 50-55, 10 April, 2006, Venice, Italia

Zanzotto, F.M., Moschitti, A. 2006. *Automatic learning of textual entailments with cross-pair similarities*. Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 401–408, Sydney, July 2006

8. Annexes

8.1. Annex 1: Example of a Parsed DLR Entry


```
<entry><list>DORITÓR, -OÁRE   1.  ◊  ◊   2. ♦ ◊    3. ◊  ◊  ◊  ♦   4.
n-14</list>

<sense class="0" value="DORITÓR, -OÁRE ">

<definition><MorfDef>adj., s. f. </MorfDef></definition>

<sense class="6" value=" 1. ">

<definition><MorfDef>Adj. </MorfDef><RegDef>Care simte plăcere,
bucurie (pentru ceva), care manifestă dragoste, ataşament (faţă de
ceva) şi îndemnul de a-l vedea, de a-l avea, de a-l cultiva, de a
intra în posesia lui etc.; care aşteaptă cu drag şi nerăbdare (ceva),
care tânjeşte (după ceva); dornic (  1 ), (învechit şi popular) dorit
( I 1 ). </RegDef><DefExemList><DefExem>Români[i] din Transilvania
totdeauna au fost doritori de limba neamului. mumuleanu, c. 76/1.
Chear soldaţii erea doritori de libertatea patrii[i]
lor.</DefExem><SG><AUTHCITE source="M. R." author="CĂPĂŢINEANU"
sigla="CĂPĂŢINEANU, M. R." pages="125/11">CĂPĂŢINEANU, M. R.
125/11</AUTHCITE>. </SG></DefExemList><DefExemList><DefExem>Domnitor
al ţării aceştia ieste Măria Sa Alecsandru Dimitrie Ghica: om foarte
blajin ... doritor fierbinte de înflorirea patrii[i].
</DefExem><SG><AUTHCITE source="G." author="TÂMPEANUL"
sigla="TÂMPEANUL, G." pages="15/22">TÂMPEANUL, G. 15/22</AUTHCITE>.
</SG><DefExem>Să cercetăm şi să întrebăm şi pe această naţie română
atât doritoare astăzi de viaţă, ce a făcut? </DefExem><SG><AUTHCITE
source="M. V." author="BĂLCESCU" sigla="BĂLCESCU, M. V."
pages="5">BĂLCESCU, M. V. 5</AUTHCITE>. </SG><DefExem>Publicul e
doritor de a vedea ziua renaşterii a literaturii naţionale (a.
1852).</DefExem><SG><SRCCITE source="PLR" volume="I" pages="150">PLR
I, 150</SRCCITE>. </SG><DefExem>Aştept, doritor, răspuns amabil.
</DefExem><SG><AUTHCITE source="O." volume="VII" author="CARAGIALE"
sigla="CARAGIALE, O." pages="79">CARAGIALE, O. VII, 79</AUTHCITE>.
</SG><DefExem>Mor în străinătatea europeană, spre care-i chiamă
gusturile lor de oameni bogaţi şi doritori de o viaţă plăcută.
</DefExem><SG><AUTHCITE source="C. I." volume="I" author="IORGA"
sigla="IORGA, C. I." pages="61">IORGA, C. I. I, 61</AUTHCITE>.
</SG><DefExem>Un nou Eminescu apăru: minte setoasă de a şti, suflet
doritor de a se împărtăşi altora. </DefExem><SG><AUTHCITE source="P.
A." volume="II" author="id." pages="130">id. P. A. II, 130</AUTHCITE>.
</SG><DefExem>Şi mai află că Laura este doritoare de-a te vedea în
casa noastră cât mai grabnic. </DefExem><SG><AUTHCITE source="I."
author="REBREANU" sigla="REBREANU, I." pages="113">REBREANU, I.
113</AUTHCITE>. </SG><DefExem>Oamenii doritori de libertate pribegeau
ori peste graniţă, ori prin codri în haiducie. </DefExem><SG><AUTHCITE
source="E." author="SADOVEANU" sigla="SADOVEANU, E."
pages="24">SADOVEANU, E. 24</AUTHCITE>. </SG><DefExem>Eram curios s-o
văd, dar nu doritor. </DefExem><SG><AUTHCITE source="I."
```

author="PREDA" sigla="PREDA, I." pages="119">PREDA, I. 119</AUTHCITE>. </SG><DefExem>Iară omul cel doritoriu de viaţă lungă stătu înaintea celui prea înalt şi cerşu şi aceşti 30 de ani. </DefExem><SG><SRCCITE source="POP.">POP.</SRCCITE>, ap. GCR II, 358. </SG></DefExemList></definition>

<sense class="10" value="◊">

<definition><SpecDef>(Substantivat)</SpecDef><DefExemList><DefExem>Ce, tu, părinte preacinstite, să ne erţi greşala carii te chemăm, cela ce-nvăţai tuturora pocăinţă şi, ca cuconilor iubitorilor de părinte, dă-te noă, şi pre doritorii tăi, cu venirea ta, veseleaşte. </DefExem><SG><AUTHCITE source="V. S." author="DOSOFTEI" sigla="DOSOFTEI, V. S.">DOSOFTEI, V. S.</AUTHCITE> ianuarie 37{sup}r{/sup}/9. </SG><DefExem> Rugăm pre Preaputernicul Creator să vă păzască în perfectă sănătate şi fericire, ca pre un doritor şi făcător de bine al patriei noastre (a. 1774).</DefExem><SG><SRCCITE source="URICARIUL" volume="I" pages="177">URICARIUL, I, 177</SRCCITE>. </SG><DefExem>Toţi cei doritori de folosul şi înflorirea patrii[i] s-au însoţit pentru un acest sfârşit şi au ajutorat cu ceia ce s-au putut </DefExem><SG>(a. 1830). <SRCCITE source="DOC. EC." pages="470">DOC. EC. 470</SRCCITE>. </SG></DefExemList></definition>

</sense>

<sense class="10" value="◊">

<definition><SpecDef>(Prin lărgirea sensului)</SpecDef><DefExemList><DefExem>Toată cugetarea ei se-mprospătase, toate visele ei reveneau splendide şi doritoare de viaţă. </DefExem><SG><AUTHCITE source="P. L." author="EMINESCU" sigla="EMINESCU, P. L." pages="102">EMINESCU, P. L. 102</AUTHCITE>. </SG></DefExemList></definition>

</sense>

</sense>

<sense class="6" value="2.">

<definition><MorfDef>Adj. </MorfDef><RegDef>Care manifestă afecţiune, prietenie faţă de cineva, plin de dor (3), de duioşie pentru cineva, p r i e t e n o s (1), a f e c t u o s;   stăpânit de dorul de a revedea pe cineva, de a fi împreună; dornic (2), (învechit şi popular) dorit (II 2). </RegDef><DefExemList><DefExem>Al tău prea doritor părinte, treti-logofăt Ghinea Păturică. </DefExem><SG><AUTHCITE source="O." volume="I" author="FILIMON" sigla="FILIMON, O." pages="127">FILIMON, O. I, 127</AUTHCITE>. </SG><DefExem>Eu sunt mamă doritoare Şi trăiesc cu supărare. </DefExem><SG><SRCCITE source="FOLC. MOLD." volume="II" pages="80">FOLC. MOLD. II, 80</SRCCITE>. </SG></DefExemList></definition>

<sense class="8" value="♦">

<definition><RegDef> Care nutreşte sentimente erotice faţă de cineva şi dorul de a-l vedea, de a fi împreună, d r ă g ă s t o s, i u b i t

o r; s p e c. stăpânit de plăceri sexuale; dornic (2), (învechit şi popular) dorit (II 2). </RegDef><ItalMarker> Gură tu! învaţă minte, nu mă spune nimănui, Nici chiar lui, când vine noaptea lângă patul meu tiptil, Doritor ca o femeie şi viclean ca un copil. EMINESCU, O. I, 80. Legându-mi, doritoare, colan de braţe moi, Mă-ntrebi: „E luna nouă alunecând prin foi?” PILLAT, P. 15. Păsăruică doritoare, Ăsta-i dor din depărtare ... Deşi sunt departe dus, Dorul tău tot m-a ajuns. FOLC. MOLD. I, 573. </ItalMarker></definition>

<sense class="10" value="◊">

<definition><SpecDef>(Prin lărgirea sensului)</SpecDef><DefExemList><DefExem>Tu, fata mea, ai putut să te uiţi cu ochi doritori la acel mai despuiat de toată simţirea bună! </DefExem><SG><AUTHCITE source="O." volume="II" author="SLAVICI" sigla="SLAVICI, O." pages="143">SLAVICI, O. II, 143</AUTHCITE>. </SG><DefExem>Îi place-aşa iubirii, Să se trădeze însăşi din umbletul pornirii, Din colţul plin de zâmbet al doritoarei guri. </DefExem><SG><AUTHCITE source="S." author="COŞBUC" sigla="COŞBUC, S." pages="32">COŞBUC, S. 32</AUTHCITE>. </SG><DefExem>Aşa împăratul june ... se întoarce îndată Cu inima ne-mpăcată Şi cu gânduri doritoare Pentru rumena floare. </DefExem><SG><AUTHCITE source="P. P." author="TEODORESCU" sigla="TEODORESCU, P. P." pages="171">TEODORESCU, P. P. 171</AUTHCITE>. </SG></DefExemList></definition>

</sense>

</sense>

</sense>

<sense class="6" value="3.">

<definition><MorfDef>Adj. </MorfDef><RegDef>Care este hotărât (pentru ceva); care vrea, care intenţionează (să facă ceva); care aspiră (către ceva); care este de acord cu ceva, care consimte (la ceva); care este interesat (în ceva); care este amator (de ceva); dornic (3), (învechit şi popular) dorit (II 3). </RegDef><DefExemList><DefExem>Doritoriu de a iscodi toate, au întrebat cu ce chip calcă poama. </DefExem><SG><AUTHCITE source="R." author="DRĂGHICI" sigla="DRĂGHICI, R." pages="28/21">DRĂGHICI, R. 28/21</AUTHCITE>. </SG><DefExem>Jurnalele tutulor naţiilor satură curiositatea unui public doritor de novităţile lumei politice. </DefExem><SG><SRCCITE source="DACIA LIT." pages="135/7">DACIA LIT. 135/7</SRCCITE>. </SG><DefExem>Să dea pricină de cercetare unor bărbaţi doritori a se pune oarecare reguli de obşte. </DefExem><SG><SRCCITE source="CR" pages="131" year="1839">CR (1839), 131</SRCCITE>{sup}1{/sup}/29. </SG><DefExem>Ei sânt ca toţi lăcuitorii ţărilor calde: doritori de izbândă. </DefExem><SG><AUTHCITE source="I." volume="I" author="RUS" sigla="RUS, I." pages="120/21">RUS, I. I, 120/21</AUTHCITE>. </SG><DefExem>Un erou cu setea primejdielor de războaie, Doritor numai de slava sângelui vărsat puhoaie. </DefExem><SG><AUTHCITE source="P." author="CONACHI" sigla="CONACHI, P." pages="282">CONACHI, P. 282</AUTHCITE>. </SG><DefExem>Făcându-ţi prin aceasta scrierile lor foarte neplăcute

înaintea poporului român celui doritoriu de citire.
</DefExem><SG><AUTHCITE source="GR." author="BĂLĂŞESCU"
sigla="BĂLĂŞESCU, GR." pages="112/26">BĂLĂŞESCU, GR.
112/26</AUTHCITE>, cf. <AUTHCITE source="D." author="PONTBRIANT"
sigla="PONTBRIANT, D.">PONTBRIANT, D.</AUTHCITE>, <AUTHCITE
author="COSTINESCU" sigla="COSTINESCU">COSTINESCU</AUTHCITE>.
</SG><DefExem>Regele, doritor de a vedea mersul constituţional mai
asigurat ..., îi repetă acelaşi sfat. </DefExem><SG><AUTHCITE
source="D." volume="V" author="MAIORESCU" sigla="MAIORESCU, D."
pages="84">MAIORESCU, D. V, 84</AUTHCITE>. </SG><DefExem>O vedea atât
de cuminte, ... atât de doritoare de a intra în voile lui.
</DefExem><SG><AUTHCITE source="O." volume="II" author="SLAVICI"
sigla="SLAVICI, O." pages="50">SLAVICI, O. II, 50</AUTHCITE>.
</SG><DefExem>Eram doritor să mă fac cunoscut, să dau semne de viaţă.
</DefExem><SG><AUTHCITE source="PLR" volume="I" author="I. NEGRUZZI"
pages="241">I. NEGRUZZI, în PLR I, 241</AUTHCITE>.
</SG><DefExem>Doritor să-şi desăvârşească studiile, ar fi voit să
plece cât mai repede în Viena. </DefExem><SG><SRCCITE source="SĂM."
volume="II" pages="202">SĂM. II, 202</SRCCITE>. </SG><DefExem>Aceştia
erau să dea pilda unei limbi literare ... tinerimei doritoare de a se
instrui. </DefExem><SG><SRCCITE source="LUC." volume="II"
pages="339">LUC. II, 339</SRCCITE>. </SG><DefExem>Doritor de a şti
orice, de a se deprinde cu orice împrejurări, de a primi pe umerii săi
zdraveni orice fel de sarcină. </DefExem><SG><AUTHCITE source="P. A."
volume="II" author="IORGA" sigla="IORGA, P. A." pages="94">IORGA, P.
A. II, 94</AUTHCITE>. </SG><DefExem>Sunt foarte ... doritor a nu-ţi
face nici ţie, nici guvernului cea mai mică complicaţiune.
</DefExem><SG><AUTHCITE source="D." author="TITULESCU"
sigla="TITULESCU, D." pages="229">TITULESCU, D. 229</AUTHCITE>.
</SG><DefExem>Lângă tătuţa sunt doritor să umblu şi eu, ca să-i fiu
Domniei Sale tovărăşie. </DefExem><SG><AUTHCITE source="O."
volume="XIII" author="SADOVEANU" sigla="SADOVEANU, O."
pages="819">SADOVEANU, O. XIII, 819</AUTHCITE>. </SG><DefExem>Vedeau
în mine ... un om liniştit, doritor de a le fi de folos.
</DefExem><SG><AUTHCITE source="C." author="ULIERU" sigla="ULIERU, C."
pages="93">ULIERU, C. 93</AUTHCITE>, cf. <AUTHCITE source="D."
author="SCRIBAN" sigla="SCRIBAN, D.">SCRIBAN,
D.</AUTHCITE></SG><DefExem>Veneau acolo tineri ... doritori să se
popească. </DefExem><SG><AUTHCITE source="I. C." author="CĂLINESCU"
sigla="CĂLINESCU, I. C." pages="53">CĂLINESCU, I. C. 53</AUTHCITE>.
</SG><DefExem> Istoricul literar, doritor să se specializeze pentru
perioadele mai vechi ale evoluţiei literare, trebuie să dobândească
cunoştinţele unui paleograf. </DefExem><SG><AUTHCITE source="L. R."
author="VIANU" sigla="VIANU, L. R." pages="14">VIANU, L. R.
14</AUTHCITE>. </SG><DefExem>Doritor să găsească o clipă de
destindere, îşi lăsă mâinile în jos. </DefExem><SG><AUTHCITE
source="P." author="VORNIC" sigla="VORNIC, P." pages="53">VORNIC, P.
53</AUTHCITE>. </SG><DefExem>Totul este înregistrat cu spiritul omului
nou, doritor de cultură enciclopledică. </DefExem><SG><SRCCITE
source="IST. LIT. ROM." volume="II" pages="145">IST. LIT. ROM. II,
145</SRCCITE>. </SG><DefExem>Doritor să cunoască trecutul istoric al
ţării sale ... cerceta ... arhivele de acolo. </DefExem><SG><SRCCITE

source="MAGAZIN IST." pages="1968">MAGAZIN IST. 1968</SRCCITE>, nr. 2, 47. </SG><DefExem>Credeţi că s-ar putea strânge în jurul muzeului, un nucleu de oameni doritori să contribuie la popularizarea acţiunilor? </DefExem><SG><SRCCITE source="CONTEMP." pages="1975">CONTEMP. 1975</SRCCITE>, nr. 1 506, 5/8. </SG><DefExem>De o neastâmpărată inteligenţă, era doritor de a şti şi a citi orice. </DefExem><SG><SRCCITE source="MAGAZIN IST." pages="1975">MAGAZIN IST. 1975</SRCCITE>, nr. 4, 48. </SG></DefExemList></definition>

<sense class="10" value="◊">

<definition><SpecDef>(Substantivat)</SpecDef><DefExemList><DefExem>Doritorii de a ajuta această întreprindere vor binevoi a se subscrie şi a o trimite înapoi la redacţia acestei foi. </DefExem><SG><AUTHCITE source="PLR" volume="I" author="HELIADE" pages="25">HELIADE, în PLR I, 25</AUTHCITE>. </SG><DefExem>Doritorii de a avea această carte se vor adresa ... la redacţia aceştii foi. </DefExem><SG><SRCCITE source="FM" pages="176" year="1843">FM (1843), 176</SRCCITE>{sup}2{/sup}/28. </SG><DefExem>Capela imperială ... o cercetai, fiind pururea deschisă doritorilor de a o vedea. </DefExem><SG><AUTHCITE source="C." author="CODRU-DRĂGUŞANU" sigla="CODRU-DRĂGUŞANU, C." pages="196">CODRU-DRĂGUŞANU, C. 196</AUTHCITE>. </SG><DefExem>Mulţumind pe deplin curiositatea cea filologhică a doritorilor. </DefExem><SG><AUTHCITE source="GR." author="BĂLĂŞESCU" sigla="BĂLĂŞESCU, GR.">BĂLĂŞESCU, GR.</AUTHCITE> III/30. </SG><DefExem>Aceşti curtizani, marii şi mărunţii doritori de a stăpâni ... nu vor lipsi. </DefExem><SG><AUTHCITE source="O." volume="III" author="CARAGIALE" sigla="CARAGIALE, O." pages="101">CARAGIALE, O. III, 101</AUTHCITE>. </SG><DefExem>An de an năvăliau în Moldova cazacii ... ajutându-se cu doritorii de domnie. </DefExem><SG><AUTHCITE source="C. I." volume="II" author="IORGA" sigla="IORGA, C. I." pages="155">IORGA, C. I. II, 155</AUTHCITE>, cf. <AUTHCITE source="D." author="RESMERIŢĂ" sigla="RESMERIŢĂ, D.">RESMERIŢĂ, D.</AUTHCITE>, CADE. </SG><DefExem>Povesti celor doritori de amănunte că primul-ministru a avut adineaori o audienţă la rege. </DefExem><SG><AUTHCITE source="R." volume="II" author="REBREANU" sigla="REBREANU, R." pages="78">REBREANU, R. II, 78</AUTHCITE>. </SG><DefExem>Trecură … dând prilej doritorilor să-şi aleagă favoritul. </DefExem><SG><AUTHCITE source="P." author="TUDORAN" sigla="TUDORAN, P." pages="507">TUDORAN, P. 507</AUTHCITE>. </SG><DefExem>Doritorii de informaţii au înroşit telefoanele. </DefExem><SG><SRCCITE source="RL" pages="2007">RL 2007</SRCCITE>, nr. 5 270. </SG></DefExemList></definition>

</sense>

<sense class="10" value="◊">

<definition><SpSpecDef> F i g . </SpSpecDef><DefExemList><DefExem>Melodiile ... se scurg în afară doritoare de-a trezi alte suflete, de-a emoţiona alte inemi. </DefExem><SG><SRCCITE source="CONTEMPORANUL">CONTEMPORANUL</SRCCITE>, VII{sub}2{/sub}, 326. </SG><DefExem>Din crengi lucesc frumoasele podoabe Ca doritoare să le facem roabe. </DefExem><SG><SRCCITE

source="SĂM." volume="II" pages="446">SĂM. II, 446</SRCCITE>.
</SG></DefExemList></definition>

</sense>

<sense class="10" value="◊">

<definition><SpecDef>(Prin lărgirea
sensului)</SpecDef><DefExemList><DefExem>Împăratul să apucă de arme,
fără de personalnica rugare, din lipsele ceale duritoare, din cea
neîmpotrivitaore rugătoare deatorniţă.</DefExem><SG><SRCCITE
source="MANIFEST" pages="23/19" year="1813">MANIFEST (1813),
23/19</SRCCITE>. </SG><DefExem>V-am deprins a vă lipsi de feliuri de
gusturi mult doritoare, fără a vă măhni. </DefExem><SG><AUTHCITE
source="R." author="DRĂGHICI" sigla="DRĂGHICI, R."
pages="111/10">DRĂGHICI, R. 111/10</AUTHCITE>. </SG><DefExem>Duhul
tău, care degeaba se luptă şi se munceşte, Doritor tot de aflare pe
cât viaţa îl hrăneşte. </DefExem><SG><AUTHCITE source="P."
author="CONACHI" sigla="CONACHI, P." pages="276">CONACHI, P.
276</AUTHCITE>. </SG><DefExem>Gâtul nalt lua acea energie marmoree şi
doritoare totodată. </DefExem><SG><AUTHCITE source="P. L."
author="EMINESCU" sigla="EMINESCU, P. L." pages="77">EMINESCU, P. L.
77</AUTHCITE>. </SG><DefExem>Vorbe şoptite la urechi doritoare de aşa
ceva. </DefExem><SG><AUTHCITE source="PRINC." author="BARBU"
sigla="BARBU, PRINC." pages="46">BARBU, PRINC. 46</AUTHCITE>.
</SG></DefExemList></definition>

</sense>

<sense class="8" value="♦">

<definition><SpecDef>(Gram.; învechit, rar)</SpecDef><RegDef> Care
arată, care exprimă o dorinţă ( ). </RegDef><ItalMarker> Chipul
furmei cei proaste de câte feliuri este? De doao feliuri: s i l i t
o r i şi d o r i t o r i ... Ce este cel doritori? Este prin care
poftim orice lucrare şi să sfârşeşte la „-z”, precum: „vănez”,
„îndemnez”. EUSTATIEVICI, GR. RUM.{sup}2{/sup} 49.
</ItalMarker></definition>

</sense>

</sense>

<sense class="6" value="4. ">

<definition><MorfDef>S. f. </MorfDef><SpecDef>(Bot.)</SpecDef><RegDef>
Plantă erbacee din familia scrofulariacee, cu tulpinile întinse pe
pământ, cu flori mici, de culoare albastru-deschis sau violet,
solitare la baza frunzei (Veronica hederifolia).
</RegDef><SG>Cf.<AUTHCITE source="PL." author="PANŢU" sigla="PANŢU,
PL.">PANŢU, PL.</AUTHCITE>, <AUTHCITE source="D." author="BORZA"
sigla="BORZA, D." pages="178, 229">BORZA, D. 178, 229</AUTHCITE>.
</SG></definition>

</sense>

</sense>

```
<MorfologicalPart><p>- Pl.: <i>doritor, -oare. </i>- Şi: (învechit)
<b>doritóriu, -ie, duritór, -oáre </b>adj. </p><p>-
<b>Dori{sup}2{/sup}{sub} {/sub}</b>+ suf. -
<i>tor.</i></p></MorfologicalPart>

</entry>
```


8.2. Annex 2: VerbNet Class “*eat-39.1*”


```
<VNCLASS ID="eat-39.1" xsi:noNamespaceSchemaLocation="vn_schema-
3.xsd">

<MEMBERS/>

<THEMROLES>

<THEMROLE type="Agent">

<SELRESTRS><SELRESTR Value="+" type="animate"/></SELRESTRS>

</THEMROLE>

<THEMROLE type="Source">

<SELRESTRS/>

</THEMROLE>

</THEMROLES>

<FRAMES/>

<SUBCLASSES>

<VNSUBCLASS ID="eat-39.1-1">

<MEMBERS><MEMBER name="eat" wn="eat%2:34:00 eat%2:34:01 eat%2:34:02"
grouping="eat.01"/>

</MEMBERS>

<THEMROLES>

<THEMROLE type="Patient">

<SELRESTRS><SELRESTR Value="+" type="comestible"/><SELRESTR Value="+"
type="solid"/>

</SELRESTRS>

</THEMROLE>

</THEMROLES>

<FRAMES>

<FRAME><DESCRIPTION descriptionNumber="" primary="NP V NP"
secondary="Basic Transitive" xtag=""/><EXAMPLES><EXAMPLE>Cynthia ate
the peach.</EXAMPLE></EXAMPLES><SYNTAX><NP
value="Agent"><SYNRESTRS/></NP><VERB/><NP
value="Patient"><SYNRESTRS/></NP></SYNTAX><SEMANTICS><PRED
value="take_in"><ARGS><ARG type="Event" value="during(E)"/><ARG
type="ThemRole" value="Agent"/><ARG type="ThemRole"
value="Patient"/></ARGS></PRED></SEMANTICS></FRAME>
```

<FRAME><DESCRIPTION descriptionNumber="" primary="NP V" secondary="Unspecified Object" xtag=""/><EXAMPLES><EXAMPLE>Cynthia ate.</EXAMPLE></EXAMPLES><SYNTAX><NP value="Agent"><SYNRESTRS/></NP><VERB/></SYNTAX><SEMANTICS><PRED value="take_in"><ARGS><ARG type="Event" value="during(E)"/><ARG type="ThemRole" value="Agent"/><ARG type="ThemRole" value="?Patient"/></ARGS></PRED></SEMANTICS></FRAME>

<FRAME><DESCRIPTION descriptionNumber="" primary="NP V PP-Conative" secondary="Conative" xtag=""/><EXAMPLES><EXAMPLE>Cynthia ate at the peach.</EXAMPLE></EXAMPLES><SYNTAX><NP value="Agent"><SYNRESTRS/></NP><VERB/><PREP value="at"><SELRESTRS/></PREP><NP value="Patient"><SYNRESTRS/></NP></SYNTAX><SEMANTICS><PRED value="take_in"><ARGS><ARG type="Event" value="during(E)"/><ARG type="ThemRole" value="Agent"/><ARG type="ThemRole" value="Patient"/></ARGS></PRED></SEMANTICS></FRAME>

<FRAME><DESCRIPTION descriptionNumber="" primary="NP V NP ADJ" secondary="NP-ADJPResultative" xtag=""/><EXAMPLES><EXAMPLE>Cynthia ate herself sick.</EXAMPLE></EXAMPLES><SYNTAX><NP value="Agent"><SYNRESTRS/></NP><VERB/><NP value="Oblique"><SELRESTRS><SELRESTR Value="+" type="refl"/></SELRESTRS></NP><ADJ/></SYNTAX><SEMANTICS><PRED value="take_in"><ARGS><ARG type="Event" value="during(E)"/><ARG type="ThemRole" value="Agent"/><ARG type="ThemRole" value="?Patient"/></ARGS></PRED><PRED value="Pred"><ARGS><ARG type="Event" value="result(E)"/><ARG type="ThemRole" value="Oblique"/></ARGS></PRED></SEMANTICS></FRAME>

<FRAME><DESCRIPTION descriptionNumber="" primary="NP V PP.source" secondary="PPSource-PP" xtag=""/><EXAMPLES><EXAMPLE>He ate off of the table.</EXAMPLE></EXAMPLES><SYNTAX><NP value="Agent"><SYNRESTRS/></NP><VERB/><PREP><SELRESTRS><SELRESTR Value="+" type="src"/></SELRESTRS></PREP><NP value="Source"><SYNRESTRS/></NP></SYNTAX><SEMANTICS><PRED value="take_in"><ARGS><ARG type="Event" value="during(E)"/><ARG type="ThemRole" value="Agent"/><ARG type="ThemRole" value="?Patient"/></ARGS></PRED><PRED value="Prep"><ARGS><ARG type="Event" value="during(E)"/><ARG type="ThemRole" value="?Patient"/><ARG type="ThemRole" value="Source"/></ARGS></PRED></SEMANTICS></FRAME></FRAMES><SUBCLASSES />

</VNSUBCLASS>

<VNSUBCLASS ID="eat-39.1-2"><MEMBERS><MEMBER name="drink" wn="drink%2:34:00 drink%2:34:01 drink%2:34:12" grouping="drink.01 drink.04"/></MEMBERS>

<THEMROLES>

<THEMROLE type="Patient"><SELRESTRS><SELRESTR Value="+" type="comestible"/><SELRESTR Value="-" type="solid"/></SELRESTRS>

</THEMROLE>

```
</THEMROLES>

<FRAMES>

<FRAME><DESCRIPTION descriptionNumber="" primary="NP V NP"
secondary="Basic Transitive" xtag=""/><EXAMPLES><EXAMPLE>Cynthia drank
the wine.</EXAMPLE></EXAMPLES><SYNTAX><NP
value="Agent"><SYNRESTRS/></NP><VERB/><NP
value="Patient"><SYNRESTRS/></NP></SYNTAX><SEMANTICS><PRED
value="take_in"><ARGS><ARG type="Event" value="during(E)"/><ARG
type="ThemRole" value="Agent"/><ARG type="ThemRole"
value="Patient"/></ARGS></PRED></SEMANTICS></FRAME>

<FRAME><DESCRIPTION descriptionNumber="" primary="NP V"
secondary="Unspecified Object" xtag=""/><EXAMPLES><EXAMPLE>Cynthia
drank.</EXAMPLE></EXAMPLES><SYNTAX><NP
value="Agent"><SYNRESTRS/></NP><VERB/></SYNTAX><SEMANTICS><PRED
value="take_in"><ARGS><ARG type="Event" value="during(E)"/><ARG
type="ThemRole" value="Agent"/><ARG type="ThemRole"
value="?Patient"/></ARGS></PRED></SEMANTICS></FRAME>

<FRAME><DESCRIPTION descriptionNumber="" primary="NP V NP ADJ"
secondary="NP-ADJPResultative" xtag=""/><EXAMPLES><EXAMPLE>Cythia
drank herself sick.</EXAMPLE></EXAMPLES><SYNTAX><NP
value="Agent"><SYNRESTRS/></NP><VERB/><NP
value="Oblique"><SELRESTRS><SELRESTR Value="+"
type="refl"/></SELRESTRS></NP><ADJ/></SYNTAX><SEMANTICS><PRED
value="take_in"><ARGS><ARG type="Event" value="during(E)"/><ARG
type="ThemRole" value="Agent"/><ARG type="ThemRole"
value="?Patient"/></ARGS></PRED><PRED value="Pred"><ARGS><ARG
type="Event" value="result(E)"/><ARG type="ThemRole"
value="Oblique"/></ARGS></PRED></SEMANTICS></FRAME>

<FRAME><DESCRIPTION descriptionNumber="" primary="NP V PP.source"
secondary="PPSource-PP" xtag=""/><EXAMPLES><EXAMPLE>He drank out of
the goblet.</EXAMPLE></EXAMPLES><SYNTAX><NP
value="Agent"><SYNRESTRS/></NP><VERB/><PREP><SELRESTRS><SELRESTR
Value="+" type="src"/></SELRESTRS></PREP><NP
value="Source"><SYNRESTRS/></NP></SYNTAX><SEMANTICS><PRED
value="take_in"><ARGS><ARG type="Event" value="during(E)"/><ARG
type="ThemRole" value="Agent"/><ARG type="ThemRole"
value="?Patient"/></ARGS></PRED><PRED value="Prep"><ARGS><ARG
type="Event" value="during(E)"/><ARG type="ThemRole"
value="?Patient"/><ARG type="ThemRole"
value="Source"/></ARGS></PRED></SEMANTICS></FRAME></FRAMES><SUBCLASSES
/>

</VNSUBCLASS>

</SUBCLASSES>

</VNCLASS>
```


8.3. Annex 3: RoWN Entries for Noun “casă”


```
<SYNSET>
```

```
  <ID>ENG20-07489779-n</ID>
  <POS>n</POS>
  <SYNONYM>
    <LITERAL>casă<SENSE>1.10.2</SENSE></LITERAL>
    <LITERAL>neam<SENSE>9.4</SENSE></LITERAL>
    <LITERAL>familie<SENSE>2.3</SENSE></LITERAL>
  </SYNONYM>
  <DEF>familie aristocrată</DEF>
  <STAMP>GabiHaja</STAMP>
  <ILR>ENG20-07489070-n<TYPE>hypernym</TYPE></ILR>
  <DOMAIN>factotum</DOMAIN>
  <SUMO>FamilyGroup<TYPE>=</TYPE></SUMO>
</SYNSET>
<SYNSET>
  <ID>ENG20-07570097-n</ID>
  <POS>n</POS>
  <SYNONYM>
    <LITERAL>casă<SENSE>1.14</SENSE></LITERAL>
    <LITERAL>companie<SENSE>1.4.x</SENSE></LITERAL>
    <LITERAL>firmă<SENSE>2</SENSE></LITERAL>
  </SYNONYM>
  <DEF>Personalul unei organizaţii comerciale sau de afaceri.</DEF>
  <STAMP>Luigi</STAMP>
  <BCS>1</BCS>
  <ILR>ENG20-07571175-n<TYPE>hypernym</TYPE></ILR>
  <DOMAIN>enterprise</DOMAIN>
  <SUMO>Corporation<TYPE>=</TYPE></SUMO>
</SYNSET>
<SYNSET>
  <ID>ENG20-07506727-n</ID>
  <POS>n</POS>
  <SYNONYM>
    <LITERAL>casă<SENSE>1.11</SENSE></LITERAL>
    <LITERAL>căsătorie<SENSE>2.2</SENSE></LITERAL>
    <LITERAL>căsnicie<SENSE>1.2</SENSE></LITERAL>
```

```
      <LITERAL>menaj<SENSE>3</SENSE></LITERAL>
      <LITERAL>mariaj<SENSE>x</SENSE></LITERAL>
    </SYNONYM>
    <DEF>două persoane căsătorite</DEF>
    <STAMP>luigi</STAMP>
    <BCS>2</BCS>
    <ILR>ENG20-07488793-n<TYPE>hypernym</TYPE></ILR>
    <DOMAIN>factotum</DOMAIN>
    <SUMO>Group<TYPE>+</TYPE></SUMO>
  </SYNSET>
  <SYNSET>
    <ID>ENG20-07489070-n</ID>
    <POS>n</POS>
    <SYNONYM>
      <LITERAL>casă<SENSE>1.12</SENSE></LITERAL>
      <LITERAL>familie<SENSE>3</SENSE></LITERAL>
      <LITERAL>neam<SENSE>4.2</SENSE></LITERAL>
      <LITERAL>viţă<SENSE>9</SENSE></LITERAL>
    </SYNONYM>
    <DEF>Totalitatea persoanelor care se trag dintr-un strămoş comun
(rude de sânge); neam, descendenţă.</DEF>
    <STAMP>Catalin Mihaila</STAMP>
    <BCS>1</BCS>
    <ILR>ENG20-07610417-n<TYPE>hypernym</TYPE></ILR>
    <DOMAIN>factotum</DOMAIN>
    <SUMO>FamilyGroup<TYPE>+</TYPE></SUMO>
  </SYNSET>
  <SYNSET>
    <ID>ENG20-03413667-n</ID>
    <POS>n</POS>
    <SYNONYM>
      <LITERAL>casă<SENSE>1.1.1</SENSE></LITERAL>
    </SYNONYM>
    <DEF>construcţie destinată pentru a servi de locuinţă uneia sau
mai multor familii</DEF>
    <STAMP>Verginica</STAMP>
```

```
    <BCS>1</BCS>
    <ILR>ENG20-02809375-n<TYPE>hypernym</TYPE></ILR>
    <ILR>ENG20-03141215-n<TYPE>hypernym</TYPE></ILR>
    <DOMAIN>building_industry</DOMAIN>
    <SUMO>House<TYPE>=</TYPE></SUMO>
  </SYNSET>
```

8.4. Annex 4: VerbNet – FrameNet Mapping

```
<vncls class="10.1" vnmember="eject" fnframe="Removing" fnlexent="1124" versionID="vn1.5"/>
<vncls class="10.1" vnmember="eliminate" fnframe="Removing" fnlexent="1126" versionID="vn1.5"/>
<vncls class="10.1" vnmember="eradicate" fnframe="NA" fnlexent="" versionID="vn1.5"/>
<vncls class="10.1" vnmember="evict" fnframe="Removing" fnlexent="1131" versionID="vn1.5"/>
<vncls class="10.1" vnmember="excise" fnframe="Removing" fnlexent="1133" versionID="vn1.5"/>
<vncls class="10.1" vnmember="excommunicate" fnframe="Exclude_member" fnlexent="12576" versionID="vn1.5"/>
<vncls class="10.1" vnmember="expel" fnframe="Removing" fnlexent="1134" versionID="vn1.5"/>
<vncls class="10.1" vnmember="extirpate" fnframe="NA" fnlexent="" versionID="vn1.5"/>
<vncls class="10.1" vnmember="extract" fnframe="Removing" fnlexent="1137" versionID="vn1.5"/>
<vncls class="10.1" vnmember="extrude" fnframe="NA" fnlexent="" versionID="vn1.5"/>
<vncls class="10.1" vnmember="lop" fnframe="NA" fnlexent="" versionID="vn1.5"/>
<vncls class="10.1" vnmember="omit" fnframe="NA" fnlexent="" versionID="vn1.5"/>
<vncls class="10.1" vnmember="ostracize" fnframe="NA" fnlexent="" versionID="vn1.5"/>
<vncls class="10.1" vnmember="oust" fnframe="Removing" fnlexent="1138" versionID="vn1.5"/>
<vncls class="10.1" vnmember="partition" fnframe="DS" fnlexent="" versionID="vn1.5"/>
```

```
<vncls class="10.1" vnmember="pry" fnframe="DS" fnlexent="6455"
versionID="vn1.5"/>
<vncls class="10.1" vnmember="reap" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.1" vnmember="remove" fnframe="Removing"
fnlexent="1144" versionID="vn1.5"/>
<vncls class="10.1" vnmember="separate" fnframe="DS" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.1" vnmember="sever" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.1" vnmember="shoo" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.1" vnmember="subtract" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.1" vnmember="uproot" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.1" vnmember="winkle" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.1" vnmember="withdraw" fnframe="Removing"
fnlexent="1150" versionID="vn1.5"/>
<vncls class="10.1" vnmember="wrench" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.2" vnmember="banish" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.2" vnmember="deport" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.2" vnmember="evacuate" fnframe="Removing"
fnlexent="1129" versionID="vn1.5"/>
<vncls class="10.2" vnmember="expel" fnframe="Removing"
fnlexent="1134" versionID="vn1.5"/>
<vncls class="10.2" vnmember="extradite" fnframe="Extradition"
fnlexent="11784" versionID="vn1.5"/>
<vncls class="10.2" vnmember="recall" fnframe="DS" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.2" vnmember="remove" fnframe="Removing"
fnlexent="1144" versionID="vn1.5"/>
<vncls class="10.3-1" vnmember="clean" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.3-1" vnmember="clear" fnframe="Emptying"
fnlexent="1002" versionID="vn1.5"/>
<vncls class="10.3-1" vnmember="clear" fnframe="Removing"
fnlexent="1118" versionID="vn1.5"/>
```

<vncls class="10.3-1" vnmember="drain" fnframe="Emptying" fnlexent="1003" versionID="vn1.5"/>

<vncls class="10.3-1" vnmember="drain" fnframe="Removing" fnlexent="1123" versionID="vn1.5"/>

<vncls class="10.3-1" vnmember="empty" fnframe="Emptying" fnlexent="1004" versionID="vn1.5"/>

<vncls class="10.3-1" vnmember="empty" fnframe="Removing" fnlexent="1128" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="buff" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="distill" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="expunge" fnframe="Removing" fnlexent="1136" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="flush" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="leach" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="polish" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="purge" fnframe="Removing" fnlexent="1142" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="shave" fnframe="Removing" fnlexent="5425" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="skim" fnframe="Removing" fnlexent="1145" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="smooth" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="soak" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="strain" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="trim" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="weed" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="whisk" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="winnow" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1" vnmember="wring" fnframe="Manipulation" fnlexent="128" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="bail" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="dab" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="dust" fnframe="Removing" fnlexent="9371" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="erase" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="lick" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="pluck" fnframe="Removing" fnlexent="1139" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="prune" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="rinse" fnframe="Removing" fnlexent="5428" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="rub" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="scour" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="scrape" fnframe="Removing" fnlexent="15073" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="scratch" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="scrub" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="squeeze" fnframe="Manipulation" fnlexent="125" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="strip" fnframe="Removing" fnlexent="1147" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="suck" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="suction" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="swab" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="sweep" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="wash" fnframe="Removing" fnlexent="5373" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="wear" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.1-1" vnmember="wipe" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2" vnmember="file" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2" vnmember="filter" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2" vnmember="hoover" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2" vnmember="iron" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2" vnmember="plough" fnframe="DS" fnlexent="5633" versionID="vn2.0"/>

<vncls class="10.4.2" vnmember="sandpaper" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2" vnmember="sponge" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2" vnmember="towel" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2-1" vnmember="brush" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2-1" vnmember="comb" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2-1" vnmember="hose" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2-1" vnmember="mop" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2-1" vnmember="plow" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2-1" vnmember="rake" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2-1" vnmember="shear" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2-1" vnmember="shovel" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2-1" vnmember="siphon" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.4.2-1" vnmember="vacuum" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="abduct" fnframe="Kidnapping" fnlexent="2457" versionID="vn1.5"/>

<vncls class="10.5" vnmember="cabbage" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="capture" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="confiscate" fnframe="Removing" fnlexent="1119" versionID="vn1.5"/>

<vncls class="10.5" vnmember="cop" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="emancipate" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="embezzle" fnframe="Theft" fnlexent="2062" versionID="vn1.5"/>

<vncls class="10.5" vnmember="extort" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="filch" fnframe="Theft" fnlexent="1962" versionID="vn1.5"/>

<vncls class="10.5" vnmember="flog" fnframe="Theft" fnlexent="15116" versionID="vn1.5"/>

<vncls class="10.5" vnmember="grab" fnframe="Taking" fnlexent="15065" versionID="vn1.5"/>

<vncls class="10.5" vnmember="hook" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="kidnap" fnframe="Kidnapping" fnlexent="2456" versionID="vn1.5"/>

<vncls class="10.5" vnmember="knock_off" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="liberate" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="lift" fnframe="Theft" fnlexent="1966" versionID="vn1.5"/>

<vncls class="10.5" vnmember="misappropriate" fnframe="Theft" fnlexent="2063" versionID="vn2.0"/>

<vncls class="10.5" vnmember="nab" fnframe="Kidnapping" fnlexent="2459" versionID="vn1.5"/>

<vncls class="10.5" vnmember="nobble" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="pilfer" fnframe="Theft" fnlexent="1964" versionID="vn1.5"/>

<vncls class="10.5" vnmember="pinch" fnframe="Theft" fnlexent="1967" versionID="vn1.5"/>

<vncls class="10.5" vnmember="pirate" fnframe="Piracy" fnlexent="4805" versionID="vn1.5"/>

<vncls class="10.5" vnmember="plagiarize" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="purloin" fnframe="Theft" fnlexent="1961" versionID="vn1.5"/>

<vncls class="10.5" vnmember="rustle" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="seize" fnframe="Taking" fnlexent="7535" versionID="vn1.5"/>

<vncls class="10.5" vnmember="sequester" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="smuggle" fnframe="Smuggling" fnlexent="3041" versionID="vn1.5"/>

<vncls class="10.5" vnmember="snatch" fnframe="Kidnapping" fnlexent="2460" versionID="vn1.5"/>

<vncls class="10.5" vnmember="snatch" fnframe="Removing" fnlexent="1146" versionID="vn1.5"/>

<vncls class="10.5" vnmember="snatch" fnframe="Theft" fnlexent="2060" versionID="vn1.5"/>

<vncls class="10.5" vnmember="sneak" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="steal" fnframe="Theft" fnlexent="1960" versionID="vn1.5"/>

<vncls class="10.5" vnmember="swipe" fnframe="Removing" fnlexent="1148" versionID="vn1.5"/>

<vncls class="10.5" vnmember="swipe" fnframe="Theft" fnlexent="1965" versionID="vn1.5"/>

<vncls class="10.5" vnmember="take" fnframe="Removing" fnlexent="1149" versionID="vn1.5"/>

<vncls class="10.5" vnmember="thieve" fnframe="Theft" fnlexent="1968" versionID="vn1.5"/>

<vncls class="10.5" vnmember="wangle" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="weasel_out" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.5" vnmember="wrest" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="balk" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="bereave" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="bilk" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="bleed" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="break" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="cheat" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="cleanse" fnframe="Emptying" fnlexent="13652" versionID="vn1.5"/>

<vncls class="10.6" vnmember="cull" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="cure" fnframe="Cure" fnlexent="901" versionID="vn1.5"/>

<vncls class="10.6" vnmember="defraud" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="denude" fnframe="Emptying" fnlexent="4089" versionID="vn1.5"/>

<vncls class="10.6" vnmember="deplete" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="depopulate" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="deprive" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="despoil" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="disabuse" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="disarm" fnframe="Emptying" fnlexent="13111" versionID="vn1.5"/>

<vncls class="10.6" vnmember="disencumber" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="dispossess" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="divest" fnframe="Emptying" fnlexent="2477" versionID="vn1.5"/>

<vncls class="10.6" vnmember="fleece" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="gull" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="purge" fnframe="Emptying" fnlexent="1005" versionID="vn1.5"/>

<vncls class="10.6" vnmember="purge" fnframe="Removing" fnlexent="1142" versionID="vn1.5"/>

<vncls class="10.6" vnmember="purify" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="ransack" fnframe="Robbery" fnlexent="15122" versionID="vn1.5"/>

<vncls class="10.6" vnmember="relieve" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="render" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="rid" fnframe="Emptying" fnlexent="2478" versionID="vn1.5"/>

<vncls class="10.6" vnmember="rifle" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="rob" fnframe="Robbery" fnlexent="1999" versionID="vn1.5"/>

<vncls class="10.6" vnmember="sap" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="strip" fnframe="Emptying" fnlexent="1006" versionID="vn1.5"/>

<vncls class="10.6" vnmember="strip" fnframe="Removing" fnlexent="1147" versionID="vn1.5"/>

<vncls class="10.6" vnmember="unburden" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6" vnmember="wean" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6-1" vnmember="burgle" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6-1" vnmember="con" fnframe="Manipulate_into_doing" fnlexent="1897" versionID="vn1.5"/>

<vncls class="10.6-1" vnmember="drain" fnframe="Emptying" fnlexent="1003" versionID="vn1.5"/>

<vncls class="10.6-1" vnmember="drain" fnframe="Removing" fnlexent="1123" versionID="vn1.5"/>

<vncls class="10.6-1" vnmember="ease" fnframe="Cure" fnlexent="4035" versionID="vn1.5"/>

<vncls class="10.6-1" vnmember="milk" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6-1" vnmember="mulct" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6-1" vnmember="plunder" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6-1" vnmember="swindle" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6-1-1" vnmember="absolve" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6-1-1" vnmember="acquit" fnframe="Verdict" fnlexent="6180" versionID="vn1.5"/>

<vncls class="10.6-1-1" vnmember="exonerate" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6-1-1" vnmember="free" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.6-1-1" vnmember="pardon" fnframe="Pardon" fnlexent="8704" versionID="vn1.5"/>

<vncls class="10.7" vnmember="bark" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="beard" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="bone" fnframe="Emptying" fnlexent="2485" versionID="vn1.5"/>

<vncls class="10.7" vnmember="burl" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="core" fnframe="Emptying" fnlexent="2481" versionID="vn1.5"/>

<vncls class="10.7" vnmember="gill" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="gut" fnframe="Emptying" fnlexent="2483" versionID="vn1.5"/>

<vncls class="10.7" vnmember="head" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="hull" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="husk" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="lint" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="louse" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="milk" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="peel" fnframe="Emptying" fnlexent="2482" versionID="vn1.5"/>

<vncls class="10.7" vnmember="pinion" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="pip" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="pit" fnframe="Emptying" fnlexent="14827" versionID="vn1.5"/>

<vncls class="10.7" vnmember="pith" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="pod" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="poll" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="pulp" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="rind" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="scale" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="scalp" fnframe="Emptying" fnlexent="2597" versionID="vn1.5"/>

<vncls class="10.7" vnmember="seed" fnframe="Emptying" fnlexent="14828" versionID="vn1.5"/>

<vncls class="10.7" vnmember="shell" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="shuck" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="skin" fnframe="Emptying" fnlexent="2480" versionID="vn1.5"/>

<vncls class="10.7" vnmember="snail" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="stalk" fnframe="Emptying" fnlexent="14830" versionID="vn1.5"/>

<vncls class="10.7" vnmember="stem" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="stone" fnframe="Emptying" fnlexent="14829" versionID="vn1.5"/>

<vncls class="10.7" vnmember="string" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="tail" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="tassel" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="vein" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="weed" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="wind" fnframe="DS" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="worm" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.7" vnmember="zest" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="deaccent" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="debark" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="debone" fnframe="Emptying" fnlexent="13108" versionID="vn1.5"/>

<vncls class="10.8" vnmember="debowel" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="debug" fnframe="Emptying" fnlexent="2593" versionID="vn1.5"/>

<vncls class="10.8" vnmember="debur" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="declaw" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="defang" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="defeather" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="deflea" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="deflesh" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="defoam" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="defog" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="deforest" fnframe="Emptying" fnlexent="2594" versionID="vn1.5"/>

<vncls class="10.8" vnmember="defrost" fnframe="Emptying" fnlexent="2658" versionID="vn1.5"/>

<vncls class="10.8" vnmember="defuzz" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="degas" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="degerm" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="deglaze" fnframe="NA" fnlexent="" versionID="vn1.5"/>

```
<vncls class="10.8" vnmember="degrease" fnframe="Emptying"
fnlexent="2595" versionID="vn1.5"/>
<vncls class="10.8" vnmember="degrit" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="degum" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="degut" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="dehair" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="dehead" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="dehorn" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="dehull" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="dehusk" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="deice" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="deink" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="delint" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="delouse" fnframe="Emptying"
fnlexent="3871" versionID="vn1.5"/>
<vncls class="10.8" vnmember="deluster" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="demast" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="derat" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="derib" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="derind" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="desalt" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
<vncls class="10.8" vnmember="descale" fnframe="Emptying"
fnlexent="2596" versionID="vn1.5"/>
<vncls class="10.8" vnmember="desex" fnframe="NA" fnlexent=""
versionID="vn1.5"/>
```

<vncls class="10.8" vnmember="desprout" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="destarch" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="destress" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="detassel" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="detusk" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="devein" fnframe="Emptying" fnlexent="13110" versionID="vn1.5"/>

<vncls class="10.8" vnmember="dewater" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="dewax" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.8" vnmember="deworm" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.9" vnmember="mine" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.9" vnmember="quarry" fnframe="NA" fnlexent="" versionID="vn1.5"/>

<vncls class="10.10" vnmember="can" fnframe="Firing" fnlexent="5797" versionID="vn2.3"/>

<vncls class="10.10" vnmember="dismiss" fnframe="Firing" fnlexent="5796" versionID="vn2.3"/>

<vncls class="10.10" vnmember="drop" fnframe="DS" fnlexent="6250" versionID="vn2.3"/>

<vncls class="10.10" vnmember="expel" fnframe="Exclude_member" fnlexent="11171" versionID="vn2.3"/>

<vncls class="10.10" vnmember="fire" fnframe="Firing" fnlexent="5798" versionID="vn2.3"/>

<vncls class="10.10" vnmember="force_out" fnframe="NA" fnlexent="" versionID="vn2.3"/>

<vncls class="10.10" vnmember="oust" fnframe="Change_of_leadership" fnlexent="1808" versionID="vn2.3"/>

<vncls class="10.10" vnmember="remove" fnframe="DS" fnlexent="1144" versionID="vn2.3"/>

<vncls class="10.10" vnmember="sack" fnframe="Firing" fnlexent="5800" versionID="vn2.3"/>

<vncls class="10.10" vnmember="send_away" fnframe="NA" fnlexent="" versionID="vn2.3"/>

```
<vncls class="10.10" vnmember="unseat" fnframe="NA" fnlexent=""
versionID="vn2.3"/><

<vncls class="10.11" vnmember="abdicate" fnframe="NA" fnlexent=""
versionID="vn2.3"/>

<vncls class="10.11" vnmember="depart" fnframe="DS" fnlexent="983"
versionID="vn2.3"/>

<vncls class="10.11" vnmember="leave" fnframe="Quitting"
fnlexent="14569" versionID="vn2.3"/>

<vncls class="10.11" vnmember="quit" fnframe="Quitting"
fnlexent="5789" versionID="vn2.3"/>

<vncls class="10.11" vnmember="renounce" fnframe="DS" fnlexent="11046"
versionID="vn2.3"/><

<vncls class="10.11" vnmember="resign" fnframe="Quitting"
fnlexent="5790" versionID="vn2.3"/>

<vncls class="10.11" vnmember="retire" fnframe="Quitting"
fnlexent="5791" versionID="vn2.3"/>

<vncls class="10.11" vnmember="vacate" fnframe="DS" fnlexent="12904"
versionID="vn2.3"/>

<vncls class="10.11" vnmember="withdraw" fnframe="DS" fnlexent="12899"
versionID="vn2.3"/>


<vncls class="10.1" fnclass="Change_of_leadership"><roles><role
fnrole="Selector" vnrole="Agent"/><role fnrole="Old_leader"
vnrole="Theme"/><role fnrole="Old_order" vnrole="Theme"/><role
fnrole="Role" vnrole="Source"/></roles></vncls>

<vncls class="10.1" fnclass="Removing"><roles><role fnrole="Agent"
vnrole="Agent"/><role fnrole="Cause" vnrole="Agent"/><role
fnrole="Theme" vnrole="Theme"/><role fnrole="Source"
vnrole="Source"/></roles></vncls>

<vncls class="10.1" fnclass="Exclude_member"><!-- added for
Exclude_member.excommunicate.v --><roles><role fnrole="Authority"
vnrole="Agent"/><role fnrole="Member" vnrole="Theme"/><role
fnrole="Group" vnrole="Source"/></roles></vncls>

<vncls class="10.2" fnclass="Removing"><roles><role fnrole="Agent"
vnrole="Agent"/><role fnrole="Cause" vnrole="Agent"/><role
fnrole="Theme" vnrole="Theme"/><role fnrole="Source"
vnrole="Source"/><role fnrole="Goal"
vnrole="Destination"/></roles></vncls>

<vncls class="10.2" fnclass="Extradition"><!-- created for
Extradition.extradite.v --><roles><role fnrole="Authorities"
vnrole="Agent"/><role fnrole="Suspect" vnrole="Theme"/><role
fnrole="Current_jurisdiction" vnrole="Source"/><role
fnrole="Crime_jurisdiction" vnrole="Destination"/></roles></vncls>
```

<vncls class="10.3" fnclass="Emptying"><roles><role fnrole="Agent"
vnrole="Agent"/><role fnrole="Cause" vnrole="Agent"/><role
fnrole="Theme" vnrole="Theme"/><role fnrole="Source"
vnrole="Source"/></roles></vncls>

<vncls class="10.3" fnclass="Removing"><roles><role fnrole="Agent"
vnrole="Agent"/><role fnrole="Cause" vnrole="Agent"/><role
fnrole="Theme" vnrole="Theme"/><role fnrole="Source"
vnrole="Source"/></roles></vncls>

<vncls class="10.4.1" fnclass="Removing"><roles><role fnrole="Agent"
vnrole="Agent"/><role fnrole="Cause" vnrole="Agent"/><role
fnrole="Theme" vnrole="Theme"/><role fnrole="Source"
vnrole="Source"/></roles></vncls>

<vncls class="10.4.1" fnclass="Manipulation"><!-- added --
><roles><role fnrole="Agent" vnrole="Agent"/><role fnrole="Entity"
vnrole="Theme"/></roles></vncls>

<vncls class="10.5" fnclass="Removing"><roles><role fnrole="Agent"
vnrole="Agent"/><role fnrole="Cause" vnrole="Agent"/><role
fnrole="Theme" vnrole="Theme"/><role fnrole="Source"
vnrole="Source"/><role fnrole="Goal"
vnrole="Beneficiary"/></roles></vncls>

<vncls class="10.5" fnclass="Taking"><!-- added --><roles><role
fnrole="Agent" vnrole="Agent"/><role fnrole="Theme"
vnrole="Theme"/><role fnrole="Source"
vnrole="Source"/></roles></vncls>

<vncls class="10.5" fnclass="Kidnapping"><roles><role
fnrole="Perpetrator" vnrole="Agent"/><role fnrole="Victim"
vnrole="Theme"/></roles></vncls>

<vncls class="10.5" fnclass="Piracy"><roles><role fnrole="Perpetrator"
vnrole="Agent"/><role fnrole="Vehicle"
vnrole="Theme"/></roles></vncls>

<vncls class="10.5" fnclass="Smuggling"><roles><role
fnrole="Perpetrator" vnrole="Agent"/><role fnrole="Goods"
vnrole="Theme"/><role fnrole="Source" vnrole="Source"/><role
fnrole="Goal" vnrole="Beneficiary"/></roles></vncls>

<vncls class="10.5" fnclass="Theft"><roles><role fnrole="Perpetrator"
vnrole="Agent"/><role fnrole="Goods" vnrole="Theme"/><role
fnrole="Source" vnrole="Source"/><role fnrole="Victim"
vnrole="Source"/><role fnrole="Purpose" vnrole="Beneficiary"/><role
fnrole="Reason" vnrole="Beneficiary"/></roles></vncls>

<vncls class="10.6" fnclass="Cure"><roles><role fnrole="Healer"
vnrole="Agent"/><role fnrole="Affliction" vnrole="Theme"/><role
fnrole="Body_part" vnrole="Theme"/><role fnrole="Patient"
vnrole="Source"/></roles></vncls>

<vncls class="10.6" fnclass="Emptying"><roles><role fnrole="Agent"
vnrole="Agent"/><role fnrole="Cause" vnrole="Agent"/><role

```
fnrole="Theme" vnrole="Theme"/><role fnrole="Source"
vnrole="Source"/></roles></vncls>

<vncls class="10.6" fnclass="Removing"><roles><role fnrole="Agent"
vnrole="Agent"/><role fnrole="Cause" vnrole="Agent"/><role
fnrole="Theme" vnrole="Theme"/><role fnrole="Source"
vnrole="Source"/></roles></vncls>

<vncls class="10.6" fnclass="Robbery"><roles><role
fnrole="Perpetrator" vnrole="Agent"/><role fnrole="Goods"
vnrole="Theme"/><role fnrole="Victim" vnrole="Source"/><role
fnrole="Source" vnrole="Source"/></roles></vncls>

<vncls class="10.6" fnclass="Verdict"><!-- added --><roles><role
fnrole="Judge" vnrole="Agent"/><role fnrole="Charges"
vnrole="Theme"/><role fnrole="Defendant"
vnrole="Source"/></roles></vncls>

<vncls class="10.6" fnclass="Pardon"><!-- added --><roles><role
fnrole="Authority" vnrole="Agent"/><role fnrole="Offense"
vnrole="Theme"/><role fnrole="Offender"
vnrole="Source"/></roles></vncls>

<vncls class="10.6" fnclass="Manipulate_into_doing"><!-- added --
><roles><role fnrole="Manipulator" vnrole="Agent"/><role
fnrole="Goods" vnrole="Theme"/><role fnrole="Victim"
vnrole="Source"/></roles></vncls>

<vncls class="10.7" fnclass="Emptying"><roles><role fnrole="Agent"
vnrole="Agent"/><role fnrole="Cause" vnrole="Agent"/><role
fnrole="Theme" vnrole="Theme"/><role fnrole="Source"
vnrole="Source"/></roles></vncls>

<vncls class="10.8" fnclass="Emptying"><roles><role fnrole="Agent"
vnrole="Agent"/><role fnrole="Cause" vnrole="Agent"/><role
fnrole="Theme" vnrole="Theme"/><role fnrole="Source"
vnrole="Source"/></roles></vncls>

<vncls class="10.10" fnclass="Firing"><roles><role fnrole="Employer"
vnrole="Agent"/><role fnrole="Position" vnrole="Predicate"/><role
fnrole="Employee" vnrole="Theme"/><role fnrole="Task"
vnrole="Source"/><!-- add correct FE to the frame --></roles></vncls>

<vncls class="10.10" fnclass="Change_of_leadership"><roles><role
fnrole="Selector" vnrole="Agent"/><role fnrole="Role"
vnrole="Predicate"/><role fnrole="Old_leader" vnrole="Theme"/><role
fnrole="Body" vnrole="Source"/></roles></vncls>

<vncls class="10.10" fnclass="Exclude_member"><roles><role
fnrole="Authority" vnrole="Agent"/><!-- add FE for vn role Predicate
(e.g., Role) --><role fnrole="Member" vnrole="Theme"/><role
fnrole="Group" vnrole="Source"/></roles></vncls>

<vncls class="10.11" fnclass="Quitting"><roles><role fnrole="Employee"
vnrole="Agent"/><role fnrole="Position" vnrole="Source"/><role
fnrole="Employer" vnrole="Source"/></roles></vncls>
```