INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN
NATURAL LANGUAGE PROCESSING LABORATORY

WORD SENSE DISAMBIGUATION AND RECOGNIZING TEXTUAL ENTAILMENT WITH STATISTICAL METHODS

A THESIS IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN COMPUTER SCIENCE

BY:
Miguel Angel Ríos Gaona

ADVISORS:
Dr. Alexander Gelbukh
Dr. Sivaji Bandyopadhyay

MEXICO, D.F., 2010

Resumen

En esta tesis estudiamos métodos estadísticos aplicados en la solución de tareas del procesamiento del lenguaje natural. Estas tareas son: la desambiguación de sentidos de las palabras y el reconocimiento de implicación textual.

Primero presentamos una nueva medida para la asignación del sentido a una palabra ambigua basada en una modificación al algoritmo simple de Lesk. Usamos la coocurrencia de las palabras para seleccionar el sentido correcto de una palabra. Se utilizó información estadística encontrada en la Web para medida. En los datos SemCor nuestro método tiene una cobertura del 100%, lo que significa siempre da una respuesta y una precisión de 0.47. En los datos de Senseval-2, nuestra variante del método de Lesk tiene una precisión de 0.45 y supero a varios de los métodos basados en Lesk, también alcanzó una cobertura del 100%.

Finalmente propusimos una nueva medida no simétrica de causa-efecto aplicada en la tarea de reconocimiento de implicación textual. En primer lugar se realizaron búsquedas en un corpus por oraciones que contiene el marcador de discurso “porque”. Con estas oraciones se creó un conjunto de pares de causa y efecto. El reconocimiento de la implicación se basa en medir la relación causa-efecto entre el texto y la hipótesis utilizando las frecuencias relativas de las palabras de los pares de causa-efecto. En los resultados hemos superado el *baseline*, en los tres corpus de prueba del PASCAL (Reconocimiento de implicación textual, RTE).La medida muestra ser buena para determinar la clase “verdadero” y mostró ser menos precisa en la clase “falso”.

Abstract

We study statistical methods based on the use of information retrieved from the Web in attempt to solve two Natural Language Processing tasks: Word Sense Disambiguation and Recognizing Textual Entailment.

For Word Sense Disambiguation, we present a measure for semantic relatedness based on the simple Lesk algorithm. We measure kind of mutual information between the gloss of each sense of the word and the context of the word: namely, the scores of the sense *s* is the frequency (as the number of webpages found by Google) of the context where the word is substituted by the gloss of the sense *s*, divided by the frequency of the gloss itself (again, as the number of webpages found by Google). In the SemCor dataset our method has the coverage of 100% (i.e., our method always gives some answer) and an accuracy of 0.47. On the Senseval 2 dataset, our method has an accuracy of 0.45, which outperforms some other Lesk-based methods (again with 100% coverage).

For Recognizing Textual Entailment, we propose a new cause-effect non-symmetric measure. First, we search over a large corpus for sentences which contain the discourse marker “because” and create a database of cause-effect pairs. The entailment recognition is based on measuring the probability of a cause-effect relation between the Text and the Hypothesis using the relative frequencies of words from the cause-effect pairs. Our results outperform the baseline system, over the three test sets of the PASCAL Recognizing Textual Entailment Challenges (RTE). The measure is good at determining the “true” class, while it is less accurate at the “false” class.

Acknowledgements

I would like to thank my advisors, Dr. Alexander Gelbukh and Dr. Sivaji Bandyopadhyay for valuable guidance and advice.

I am also grateful to my family and friends for their understanding and support.

Table of Contents

List of Tables

List of Figures

1. Introduction

In our time most of the information is encoded in the form of natural language text. Newspapers, magazines, radio, TV and the World Wide Web (WWW) are examples of the most complex information medium in our world: human language. Therefore with these resources comes the problem of finding a specific datum in millions of documents. A human reader will take many years to do this task. Thus computers can process (less accurate than a human) the millions of available documents in few time.

The idea of giving computers the ability to process human language is as old as the idea of computers themselves (Manning and Shutze, 1999). The goal of the Natural Language Processing (NLP) is to design and build software that will analyze, understand, and generate languages that humans use naturally. This goal is not easy to reach. "Understanding" language means, among other things, knowing what concepts a word or phrase stands for and knowing how to link those concepts together in a meaningful way. Natural language system is easiest for humans to learn and use and hardest for a computer to master. Long after machines have proven capable of solving complex mathematical problems with speed and grace, they still fail to master the basics of our spoken and written languages.

Research in NLP has been going on for several decades dating back to the late 1940's. Machine translation (MT), task of translating texts from one natural language to another, was the first computer-based application related to NLP. Early work in MT took the simplistic view that the only differences between languages resided in their vocabularies and the permitted word orders. Systems developed from this perspective simply used dictionary-lookup for appropriate words for translation and reordered the words after translation to fit the word-order rules of the target language, without taking into account the lexical ambiguity inherent in natural language. Thus this produced poor results.

Natural language processing provides both theory and implementations for a range of applications. In fact, any application that utilizes text is a candidate for NLP. The most frequent applications utilizing NLP include the following:

- Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies a user information need from within large collections (usually stored on computers).
- Information Extraction (IE) focuses on the recognition, tagging, and extraction into a structured representation, certain key elements of information, e.g. persons, companies, locations, organizations, from large collections of text. These extractions can then be utilized for a range of applications including question-answering, visualization, and data mining.
- Question Answering (QA) in contrast to Information Retrieval, which provides a list of potentially relevant documents in response to a user's query, QA provides the user with either just the text of the answer itself or answer-providing passages.
- Summarization (SUM) an implementation that reduces a larger text into a shorter, yet richly constituted abbreviated narrative representation of the original document.
- Machine Translation is the application of computers to the task of translating texts from one natural language to another.
- Dialogue Systems perhaps the omnipresent application of the future, in the systems envisioned by large providers of end-user applications. Dialogue systems, which usually focus on a narrowly defined application, e.g. your refrigerator or home sound system.

The most explanatory method for presenting what actually happens within a NLP system is by means of the "*levels of language*" approach. This is also referred to engage the complex language behavior we require of various kinds of knowledge about language:

- Phonetics and Phonology—knowledge about linguistic sounds.
- Morphology—knowledge of the meaningful components of words.
- Syntax—knowledge of the structural relationships between words.
- Semantics—knowledge of meaning.
- Pragmatics—knowledge of the relationship of meaning to the goals and intentions of the speaker.
- Discourse—knowledge about linguistic units larger than a single utterance.

1.1. Research Problems and Why they are Worthwhile Studying

Ambiguity resolution improves the quality of the solution in most NLP tasks. A word is ambiguous if it has multiple senses, i.e., alternative meanings, for example:

- An MT system translates *bill* from English to Spanish. Should it translate it as *pico* "bird jaw" or *cuenta* "invoice"?
- IR retrieves all the web pages about *cricket*, so the sport or the insect.
- QA answer the query "What is *George Miller*'s position on gun control?", so George Miller, the psychologist or congressman.

Word Sense Disambiguation (WSD) is the task of selecting the most appropriate meaning for a polysemous word based on the context in which it occurs. For example, in the phrase "*The bank down the street was robbed*", the word bank means a financial institution, while in "*The city is on the Western bank of Jordan*", this word refers to the shore of a river. WSD is an intermediate task (Volk, 2002) and as we see above it is used in many applications.

Also another fundamental phenomenon in language is the variability of a semantic expression, which the same meaning could be expressed or infer from different text. For example, the query "*What does Peugeot manufacture?*" A QA system must be able to recognize, or infer, and answer which may be expressed differently from the query. For example, from text "*Chrétien visited Peugeot's newly renovated car factory*" entails the hypothesized answer from "*Peugeot manufactures cars*".

Recognizing Textual Entailment (RTE) has been proposed as a generic task that captures major semantic inference needs across many natural language processing applications. This task is defined as a directional relationship between pair of text expressions, denoted by T -the entailing "Text" and H -the entailed "Hypothesis". We say that T entails H if the meaning of H can be inferred from the meaning of T as could typically de interpreted by people.

Moreover, many NLP tasks have strong links to entailment: in SUM, a summary should be entailed by the text; Paraphrase recognition (PP) can be seen as mutual entailment

between a text T and a hypothesis H; in IE, the extracted information should also be entailed by the text; in QA the answer obtained for one question after the IR process must be entailed by the supporting snippet of text.

In this thesis we proposed two statistical methods based on the use of information retrieved from the Web as an attempt to resolve the WSD task and the RTE task. Our WSD method outperformed most of the WSD approaches. But, our RTE method only outperforms the baseline method. Thus we developed a meta-classifier based on our method which has a competitive performance. Finally some contributions of our work are: Publications.

1.2. Research Methods in Brief

We proposed two statistical approaches based on the use of the Web as a corpus as an attempt to resolve the WSD task and the RTE task.

For the WSD task we propose a variant of the Lesk algorithm. The Lesk algorithm basically disambiguates a word by measuring the word overlap of each definition of the ambiguous word against the context of the ambiguous word. Therefore we propose a variation of this scheme. Instead of measure the word overlap we measure the frequency count of the definition of a sense and the context and choose the sense with the best score.

Most of the approaches for the RTE task consist in measure the similarity between the Text and the Hypothesis. Thus many of these measures are symmetric and the RTE task is a directional relation between the Text and the Hypothesis. So, we propose a non-symmetric similarity measure for the RTE task. Our non-symmetric measure is based on find a causal relation between the Text and the Hypothesis. We measure the causal relation from the frequency count of words from sentences with the word *because* and decide if the Text Hypothesis pair is true or false.

1.3. Goal of the Thesis

The main goal of our research is to use the Web as a corpus to develop NLP statistical approaches.

Our particular goals are:

- Propose a new WSD approach based on a variant of the Lesk algorithm.
- Propose a new RTE approach based on a non-symmetric similarity measure.

1.4. Structure of the Thesis

The thesis is organized as follow:

- Chapter 2. The related literature is shown in this section.
- Chapter 3. In this chapter we show the method, results, and a comparison with previous works of the WSD approach.
- Chapter 4. The RTE approaches, results and a comparison with previous works are showed in this chapter.
- Chapter 5. The final conclusions and future work are drawn.

2. Related Bibliography

In this chapter we first show the main approaches in NLP. Second, we show how the statistical approaches are use in WSD and finally, we show the main approaches in RTE.

NLP approaches fall roughly into four categories: symbolic, statistical, connectionist, and hybrid. Symbolic and statistical approaches have coexisted since the early days of this field. Connectionist NLP work first appeared in the 1960's.

For a long time, symbolic approaches dominated the field. In the 1980's, statistical approaches regained popularity as a result of the availability of critical computational resources and the need to deal with broad, real-world contexts. Connectionist approaches also recovered from earlier criticism by demonstrating the utility of neural networks in NLP. This section examines each of these approaches in terms of their foundations, typical techniques, differences in processing and system aspects, and their robustness, flexibility, and suitability for various tasks.

Symbolic approaches perform deep analysis of linguistic phenomena and are based on explicit representation of facts about language through well-understood knowledge representation schemes and associated algorithms. In fact, the description of the levels of language analysis in the preceding section is given from a symbolic perspective.

The primary source of evidence in symbolic systems comes from human-developed rules and lexicons. A good example of symbolic approaches is seen in logic or rule-based systems. In logic-based systems, the symbolic structure is usually in the form of logic propositions.

Manipulations of such structures are defined by inference procedures that are generally truth preserving. Rule-based systems usually consist of a set of rules, an inference engine, and a workspace or working memory. Knowledge is represented as facts or rules in the rule-base. The inference engine repeatedly selects a rule whose condition is satisfied and executes the rule.

Another example of symbolic approaches is semantic networks. First proposed by Quillian (2000) to model associative memory in psychology, semantic networks represent knowledge through a set of nodes that represent objects or concepts and the labeled links that represent relations between nodes. The pattern of connectivity reflects semantic organization, that is; highly associated concepts are directly linked whereas moderately or weakly related concepts are linked through intervening concepts. Semantic networks are widely used to represent structured knowledge and have the most connectionist flavor of the symbolic models.

Symbolic approaches have been used for a few decades in a variety of research areas and applications such as information extraction, text categorization, ambiguity resolution, and lexical acquisition. Typical techniques include: explanation-based learning, rule-based learning, inductive logic programming, decision trees, conceptual clustering, and K nearest neighbor algorithm.

Statistical approaches employ various mathematical techniques and often use large text corpora to develop approximate generalized models of linguistic phenomena based on actual examples of these phenomena provided by the text corpora without adding significant linguistic or world knowledge. In contrast to symbolic approaches, statistical approaches use observable data as the primary source of evidence. A frequently used statistical model is the Hidden Markov Model (HMM) inherited from the speech community. HMM is a finite state automaton that has a set of states with probabilities attached to transitions between states. Although outputs are visible, states themselves are not directly observable, thus "*hidden*" from external observations. Each state produces one of the observable outputs with a certain probability.

Statistical approaches have typically been used in tasks such as speech recognition, lexical acquisition, parsing, part-of-speech tagging, collocations, statistical machine translation, and statistical grammar learning, and so on.

Similar to the statistical approaches, connectionist approaches also develop generalized models from examples of linguistic phenomena. What separates connectionism from other statistical methods is that connectionist models combine statistical learning with various theories of representation - thus the connectionist representations allow transformation, inference, and manipulation of logic formulae. In addition, in connectionist systems,

linguistic models are harder to observe due to the fact that connectionist architectures are less constrained than statistical ones.

Generally speaking, a connectionist model is a network of interconnected simple processing units with knowledge stored in the weights of the connections between units. Local interactions among units can result in dynamic global behavior, which, in turn, leads to computation. Some connectionist models are called localist models, assuming that each unit represents a particular concept. For example, one unit might represent the concept "*mammal*" while another unit might represent the concept "*whale*". Relations between concepts are encoded by the weights of connections between those concepts. Knowledge in such models is spread across the network, and the connectivity between units reflects their structural relationship. Localist models are quite similar to semantic networks, but the links between units are not usually labeled as they are in semantic nets. They perform well at tasks such as word-sense disambiguation, language generation, and limited inference.

Other connectionist models are called distributed models. Unlike that in localist models, a concept in distributed models is represented as a function of simultaneous activation of multiple units. An individual unit only participates in a concept representation. These models are well suited for natural language processing tasks such as syntactic parsing, limited domain translation tasks, and associative retrieval.

To summarize, symbolic, statistical, and connectionist approaches have exhibited different characteristics, thus some problems may be better tackled with one approach while other problems by another. In some cases, for some specific tasks, one approach may prove adequate, while in other cases, the tasks can get so complex that it might not be possible to choose a single best approach.

2.1. Approaches to Word Sense Disambiguation

The problem of word sense disambiguation has been described as AI-complete, that is, a problem which can be solved only by first resolving all the difficult problems in artificial intelligence (AI), such as the representation of common sense and encyclopedic knowledge (Pradhan et al. 2007).

To address this task, different methods have been used, with various degrees of success. These methods can be classified depending on the type of knowledge they use to accomplish the task. The main statistical approaches to the WSD task are supervised and unsupervised disambiguation.

2.1.1. Supervised Methods

Supervised methods use a labelled training set to solve the task. They have been shown to be the most efficient ones (Pradhan et al. 2007). However, the lack of large sense tagged corpora limits this kind of methods, and it is difficult and expensive to create such corpora manually.

Several research projects take a supervised learning approach to WSD (Brown et al. 1991). The goal is to learn to use surrounding context to determine the sense of an ambiguous word.

Often the disambiguation accuracy is strongly affected by the size of the corpus used in the process. Typically, 1000–2500 occurrences of each word are manually tagged in order to create a corpus. From this about 75% of the occurrences are use for the training phase and the remaining 25% are use for the testing (Mihalcea and Moldovan, 1999). Corpus like interest and line were the most well studied in literature.

The Interest dataset (a corpus where each occurrence of the word interest is manually marked up with one of its 6 senses) represent the context of an ambiguous word with the part-of-speech of three words to the left and right of interest, a morphological feature indicating if interest is singular or plural, an unordered set of frequently occurring keywords that surround interest, local collocations that include interest, and verb-object syntactic relationships. A nearest-neighbour classifier was employed and achieved an accuracy of 0.87 over repeated trials using randomly training and test sets. Ng and Lee (1996), and Pedersen et al. (2000) present studies that utilize the original Bruce and Wiebe feature set and include the interest data .The first compares a range of probabilistic model selection methodologies and finds that none out perform the Naive Bayesian classifier, which attains accuracy of 0.74. The second compares a range of machine learning

algorithms and finds that a decision tree learner 0.78 and a Naïve Bayesian classifier 0.74 are most accurate.

The Line dataset (similarly, a corpus where each occurrence of the word line is marked with one of its 6 senses) was first studied by Leacock (1993). They evaluate the disambiguation accuracy of a Naive Bayesian classifier, a content vector, and a neural network. The context of an ambiguous word is represented by a bag-of-words (BoW) where the window of context is two sentences wide. When the Naive Bayesian classifier is evaluated words are not stemmed and capitalization remains. With the content vector and the neural network words are stemmed and words from a stop-list are removed. They report no significant differences in accuracy among the three approaches; the Naïve Bayesian classifier achieved an accuracy of 0.71, the content vector of 0.72, and the neural network 0.76.

This dataset was studied again by Mooney (1996), where seven different machine learning methodologies are compared. All learning algorithms represent the context of an ambiguous word using the BoW with a two sentence window of context. In these experiments words from a stop list are removed, capitalization is ignored, and words are stemmed. The two most accurate methods in this study proved to be a Naive Bayesian classifier 0.72 and a perceptron 0.71.

Recently, the Line dataset was revisited by both Towell and Voorhees (1998), and Pedersen (1997). Take an ensemble approach where the output from two neural networks is combined; one network is based on a representation of local context while the other represents topical context. The latter utilize a Naive Bayesian classifier. In both cases context is represented by a set of topical and local features. The topical features correspond to the open-class words that occur in a two sentence window of context. The local features occur within a window of context three words to the left and right of the ambiguous word and include co-occurrence features as well as the PoS of words in this window. These features are represented as local and topical BoW and PoS. (Towell and Voorhees, 1998) report an accuracy of 0.87 while (Pedersen et al. 1997) report accuracy of 0.84.

2.1.2. Unsupervised Methods

Unsupervised methods are based on unlabeled corpora. This resolves the knowledge acquisition bottleneck, at the cost of low accuracy. Unsupervised approaches often do not use any learning process; they only rely on a lexical resource, like WordNet (Miller, 1991), to carry out the WSD task.

The wide used methods are the methods based on content vectors. The content vectors approach treats the ambiguous word context as a document in IR. Therefore a vector in an n-dimensional space (n the number of words in context) it is associated to each context. Each row in the vector contains a function of the frequency of each word in the context. Even in WSD many similarity measures between vectors were proven and this measures are enough different from the measures in IR. The similarity measures between vectors are used to develop sets of the most similar vectors by means of clustering. These sets can be considered as the senses of the ambiguous word.

A main contribution to this approach is the algorithm of Schutze (1992) called context-group discrimination. The context-group discrimination algorithm is similar to the method of Brown et al. (1991) which is a supervised method. The main difference between the Shutze method and the Naïve Bayesian classifier of Gale is that the context-group discrimination algorithm first takes a random sample of the parameters to later re-estimate the parameter by an Expectation-Maximization algorithm (EM). Thus from the random sample of the parameters is taken for every context of the ambiguous word the conditional probability of that word to be used in a particular context. This categorization is used for training and the EM maximizes the similarity of the data for the given model.

Schutze (1992) proposes a method which avoids tagging each occurrence in the training corpus. Using letter fourgrams within a 1001 character window, his method first automatically clusters the words in the text, and each target word is represented by a vector; a sense is then assigned manually to each cluster, rather than to each occurrence.

Assigning a sense demands examining 10 to 20 members of each cluster, and each sense may be represented by several clusters. This method reduces the amount of manual intervention but still requires the examination of a hundred or so occurrences for each ambiguous word. More seriously, it is not clear what the senses derived from the clusters

correspond to (Pereira et al. 1993); and they are not in any case directly usable by other systems, since it is derived from the corpus itself.

Brown et al. (1991) and Gale et al. (1993) propose the use of bilingual corpora to avoid hand-tagging of training data. Their premise is that different senses of a given word often translate differently in another language (for example, *pen* in English is *stylo* in French for its writing implement sense, and *enclos* for its enclosure sense). By using a parallel aligned corpus, the translation of each occurrence of a word such as *sentence* can be used to automatically determine its sense. This method has some limitations since many ambiguities are preserved in the target language (e.g., French *souris*--English *mouse*); furthermore, the few available large-scale parallel corpora are very specialized (for example, the *Hansard Corpus* of Canadian Parliamentary debates), which skews the sense representation.

Dagan et al. (1991) and Dagan and Itai (1994) propose a similar method, but instead of a parallel corpus use two monolingual corpora and a bilingual dictionary. This solves in part the problems of availability and specificity of domain that plagues the parallel corpus approach, since monolingual corpora, including corpora from diverse domains and genres are much easier to obtain than parallel corpora.

Other methods attempt to avoid entirely the need for a tagged corpus, such as many of those cited in the section below (e.g., Yarowsky, 1992, who attacks both the tagging and data sparseness problems simultaneously). However, it is likely that, as noted for grammatical tagging (Mérialdo, 1994), even a minimal phase of supervised learning improves radically on the results of unsupervised methods. Research into means to facilitate and optimize tagging is ongoing; for example, an optimization technique called *committeebased sample selection* has been proposed (Engelson and Dagan, 1996), which, based on the observation that a substantial portion of manually tagged examples contribute little to performance, enables avoiding the tagging of examples that carry more or less the same information. Such methods are promising, although to our knowledge they have not been applied to the problem of lexical disambiguation.

2.2. Approaches to Recognizing Textual Entailment

Entailment definition in formal semantics (Chierchia & McConnell-Ginet, 2001) is the following:

A text T entails another text H if H is true in every circumstance (possible world) in which T is true.

This definition imposes a strictness that is inappropriate to many practical NLP systems. The problem is addressed by the notion of *applied textual entailment*, as defined by Dagan and Glickman (2004), which takes an empirical evaluation approach. By this definition, a text T entails a hypothesis H, if, typically, a human reading T would infer that H is most likely true. The advantages of such a perspective for NLP are: the evaluation is performed using a human gold standard, as in other NLP tasks, and at the same time, common background knowledge is assumed.

Other annotation guidelines for textual entailment, by Dagan and Glickman (2004):

- Entailment is a directional relation; hypothesis must be entail by the text and not the contrary.
- The hypothesis must be totally entail by the text and don't include parts which couldn't be inferred.
- Cases in which the infer is probable high but not with absolute certain, should be judge as true.
- The background knowledge about the world, must be typical to a normal reader of that kind of text (news domain); instead isn't acceptable the know presupposition of high specific knowledge.

Based on the applied textual entailment definition, the PASCAL Network of Excellence recently started the RTE Challenge (Dagan et al. 2005). A few samples of Text-Hypothesis (T-H) pairs from the first RTE-1 Challenge are shown bellow.

Table 2.1: T-H pairs examples

TASK	TEXT	HYPOTHESIS	ENTAILMENT
IR	iTunes software has seen strong sales in Europe.	Strong sales for iTunes in Europe.	True
PP	American Airlines began laying off hundreds of flight attendants on Tuesday, after a federal judge turned aside a union's bid to block the job losses.	American Airlines will recall hundreds of flight attendants as it steps up the number of flights it operates.	False
QA	The two suspects belong to the 30th Street gang, which became embroiled in one of the most notorious recent crimes in Mexico: a shootout at the Guadalajara airport in May, 1993, that killed Cardinal Juan Jesus Posadas Ocampo and six others.	Cardinal Juan Jesus Posadas Ocampo died in 1993.	True

From Table 2.1 we can see the main goals of the RTE. To create a dataset of T-H pairs of small text snippets, corresponding to the news domain. Examples were manually labeled for entailment. Participating systems were asked to decide for each T-H pair whether T entail H or not, giving True or False annotation as a system output. For this reason the datasets provided by the RTE Challenge organizers are intended to include typical T–H pairs that correspond to success and failure cases of actual text processing applications, dealing with tasks such as IE, IR, QA and SUM. They are divided into two balanced corpora: Development and Test datasets.

The judgments (classifications) produced by the systems were compared to the gold standard. The percentage of matching judgments provides the accuracy of the run (the fraction of correct responses). As a second measure the Confidence-Weighted Score (cws, also known as Average Precision) was computed. Judgments of the test examples were sorted by their confidence (in decreasing order).

For instance, in the RTE-1 Challenge (Dagan et al. 2004), the dataset was collected from different text processing application like IR, Comparable Documents (CD), Reading Comprehension (RC), QA, IE, Machine Translation (MT) and PP. The collected examples represent a range of different levels of entailment reasoning, based on lexical syntactic,

logical and world knowledge. In the end 567 examples were in the development dataset and 800 in the test dataset, split into True/False examples.

The main focus for the RTE-2 Challenge dataset was to provide more “realistic” T-H pairs. Dataset consist of 1600 T-H pairs divided into development and test datasets, each one containing 800 pairs. The organizers focused on four applications IR, IE, QA and SUM.

RTE-3 Challenge followed the same structure of the previous versions. Something new was introduced, a resource pool, where participants had the possibility to share the same recourses.

In 2008 the RTE-4 Challenge include the three-way decision of “YES”, “NO” and “UNKNOWN” to drive systems to make more precise informational distinctions; a hypothesis being unknown on the basis of a text should be distinguished from a hypothesis being shown false/contradicted by a text. The classic two-way RTE task was also offered, in which the pairs where T entailed H were marked as ENTAILMENT, and those where the entailment did not hold were marked as NO ENTAILMENT. The descriptions of the tasks are presented below:

The three-way RTE task is to decide whether:

- T entails H - in which case the pair will be marked as ENTAILMENT.
- T contradicts H - in which case the pair will be marked as CONTRADICTION.
- The truth of H cannot be determined on the basis of T - in which case the pair will be marked as UNKNOWN.

The two-way RTE task is to decide whether:

- T entails H - in which case the pair will be marked as ENTAILMENT.
- T does not entail H - in which case the pair will be marked as NO ENTAILMENT.
- The RTE-4 dataset was made of 1000 pairs (300 each for IE and IR, 200 each for SUM and QA).

2.2.1. Approaches by Language Levels

The RTE approaches can be classified depending in which textual entailment phenomena address or the type of representation (*levels of language*) of the T-H pair.

Vanderwende et al. (2005) examine the complete test set of RTE-1 with the purpose of isolating the pairs whose categorization can be accurately predicted based only on syntactic matching. The human annotation indicates that 37% of the entailments are decided merely at the syntactic level, this outperforms to 49% if the information of a general-purpose thesaurus is additionally added.

Bar-Haim et al. (2006) take this idea a step further and annotate 30% of the RTE-1 test set at two strictly defined levels of entailment. Extending Vanderwende et al.'s work, they consider a lexical entailment level, which involves morphological derivations, ontological relations and lexical world knowledge, in addition to a lexical-syntactic level, which, on top of lexical transformations, contains syntactic transformations, paraphrases and coreference. Where the T-H pairs are decided 44% for the lexical and of 50% for the lexical-syntactic level. Clark et al. (2007) explore the requirements of RTE in a way that differs from the previous approaches in that it is not centered on the basic lexical-syntactic levels of entailment, but instead it investigates a wide range of phenomena involving lexical and world knowledge. Clark et al. manually annotate 25% of the positive entailment pairs in RTE-3 for thirteen distinct entailment phenomena. As we can see there are various levels of the entailment phenomena, classified under four main categories lexical, syntactic, semantic and logical.

2.2.1.1. Lexical Level

The approach in (Pérez and Alfonseca, 2005) consists in using the BLEU algorithm that works at the lexical level, to compareT-H pairs. Next, the entailment is judged as true or false according to BLEU's output. Once the algorithm is applied, they had seen that the results confirm the use of BLEU as baseline for the automatic recognition of textual entailments. They showed that a shallow technique can reach around 50% of accuracy. In

order to recognize entailments using BLEU, the first decision is to choose whether the candidate text should be considered as part of the entailment (T) or as the hypothesis (H). In order to make this choice, they did a first experiment in which they considered the T part as the reference and the H as the candidate. This setting has the advantage that the T part is usually longer than the H part and thus the reference would contain more information that the candidate.

2.2.1.2. Syntactic Level

Graph distance/similarity measures are widely recognized to be powerful tools for matching problems and it was used with success in RTE-1 by Pazienza, Pennacchiotti and Zanzotto. Objects to be matched (two images, patterns, text and hypothesis in RTE task, etc.) are represented as graphs, turning the recognition problem into a graph matching task.

Following (Dagan and Glickman, 2004), since the hypothesis H and text T may be represented by two syntactic graphs, the textual entailment recognition problem can be reduced to graph similarity measure estimation, although textual entailment has particular properties (Pazienza et al., 2005):

- Classical graph problems, it is non-symmetric.
- Node similarity can not be reduced to the label level (token similarity).
- Similarity should be estimated also considering linguistically motivated graph transformations (nominalization and passivization).

The tree edit distance algorithm (Kouylekov and Magnini, 2005) applied on the dependency trees of both the text and the hypothesis. If the distance (cost of the editing operations) among the two trees is below a certain threshold, empirically estimated on the training data, then we assign an entailment relation between the two texts. According to the approach described above, the following transformations are allowed:

- 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.

2.2.1.3. Semantic Level

The system proposed in (Bar-Haim et al. 2006) relies on a relatively deep linguistic analysis, which we complement with a shallow component based on word overlap.

The system is based on three main components:

- A linguistic analysis of text and hypothesis based primarily on LFG and Frame Semantics (Baker et al. 1998).
- A computation of a match graph that encodes the semantic overlap between text and hypothesis.
- A statistical entailment decision (Bar-Haim et al. 2006).

Bos and Markert (2005) used several shallow surface features to model the text, hypothesis and their relation to each other. They expected some dependency between the surface string similarity of text and hypothesis and the existence of entailment. This string similarity measure uses only a form of extended word overlap between text and hypothesis, taking into account identity of words, as well as synonymy and morphological derivations revealed by WordNet (Fellbaum, 1998).

To introduce an element of robustness into their approach, they used model builders to measure the “distance” from an entailment. The intuition behind this approach is as follows: 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. Put differently, the domain size for T+H would equal the domain size of T. In contrast, if T does not entail H, H normally introduces some new information (except when it contains negated information), and this will be

reflected in the domain size of T+H, which becomes larger than the domain size of T. It turns out that this difference between domain sizes is a useful way of measuring the likelihood of entailment. Large differences are mostly not entailments, small differences usually are.

They use a robust wide-coverage CCG-parser (Bos et al. 2004) to generate fine-grained semantic representations for each T-H pair. The semantic representation language is a first-order fragment used in Discourse Representation Theory (DRS) (Kamp and Reyle, 1993); including the recursive DRS structure to cover negation, disjunction, and implication. Given a T-H pair, a theorem prover can be used to find answers to the following conjectures:

- T implies H (shows entailment).
- T+H are inconsistent (shows no entailment).

In the RTE-1, five groups used logical provers and offered deep semantic analysis. One system (Raina et al. 2005) transformed the text and hypothesis into logical formula like in Harabagiu et al. (2000) and it calculated the "cost" of proving hypothesis from text. In RTE-2 only two systems used logical inferences and one of the systems achieved the second result of the edition (Tatu et al. 2006). In RTE-3 the number of systems using logical inferences grew up to seven and the first two results used the logical inferences (Hickl 2007 and Tatu 2007). In RTE-4 nine groups used logical inferences in order to identify the entailment relation, and two of them were oriented to that (Clark and Harrison, 2008) and (Bergmair, 2008).

2.2.2. Machine Learning Approaches

In the RTE-1, the number of systems that used machine learning algorithms to determine the result of the entailment relation was considerable. The aim was to use results offered by these algorithms for answer classification instead of using thresholds established by human experts on training data. The features used by these systems include lexical, semantic, grammatical attributes of verbs, nouns and adjectives, named entities, and were calculated using the WordNet taxonomy, the VerbOcean semantic network (Chlonsky and Pantel,

2004), a Latent Semantic Indexing technique (Deerwester et al. 1990), or the ROUGE metrics (Lin and Hovy, 2003). Other features like negation were identified by inspecting the semantic representation of text with DRS for the presence of negation operators.

These parameters were evaluated by machine learning algorithms such as SVM (Joachims, 2002) or such as C5.0 (Quinlan, 2000), or used binary classifications like Bayesian Logistic Regression (BBR) and TiMBL (Daelemans et al. 1998). Starting with RTE-2, the interest for using machine learning grew constantly. Thus, the number of systems that used machine learning for classification was increased from seven in RTE-1 to fifteen in RTE-2 and sixteen in RTE-3 and RTE-4. The approaches are various and their results depend on identify relevant features. In (Inkpen et al. 2006), matching features are represented by lexical matches (including synonyms and related words), part-of-speech matching and matching of grammatical dependency relations. Mismatch features include negation and numeric mismatches. The MLEnt system (Kozareva, 2006) models lexical and semantic information in the form of attributes and, based on them, proposed 17 features. In (Ferrés and Rodríguez, 2007), the authors computed a set of semantic based distances between sentences. The system of Montejo-Ráez et al. (2007) used semantic distance between stems, subsequences of consecutive stems and trigrams matching. The features identified in (Li et al. 2007) include lexical semantic similarity, named entities, dependent content word pairs, average distance, negation, and text length.

3. Word Sense Disambiguation

In this chapter we first present a new measure for semantic relatedness based on the simple Lesk algorithm. We use word co-occurrences for disambiguate the ambiguous word. The statistical information for the measure is retrieved from the Web. Our experiments are as follows: first over the Semcor corpus and then a comparison with previous results over the Senseval 2 corpus. Finally partial conclusions are drawn.

An example of an unsupervised method is the original Lesk algorithm (OL) (Lesk, 1986) that disambiguates polysemous words in (shorts) phrases. The definition, or gloss (from a dictionary), of each sense of an ambiguous word in a phrase is compared to the glosses of every other word in the phrase. Basically, the algorithm selects the set of senses such that their glosses have the largest number of words in common.

To tackle the problem of knowledge acquisition bottleneck in supervised methods, the Web could be use as a lexical resource.

The Web has become a source of data for NLP, and WSD is no an exception. Many methods use the Web to automatically generate sense tagged corpora (Martinez, 2003).

Web as a corpus for NLP research (Volk, 2002) was already used with success in many areas such as question answering (Brill et al. 2001), machine translation (Greffenstete, 1999), and anaphora resolution (Bunescu, 2003).

3.1. Theoretical Framework

Senseval, started in 1998 (Kilgarriff and Rosenzweig, 1998), tied to the evaluation of WSD systems, producing a set of benchmarks for evaluating WSD system performance, to establish the viability of WSD as a separately evaluable NLP task.

In the past versions of Senseval, exercises that were variants of the Lesk approach were considered as baseline approaches. In Senseval 1, most of the systems for disambiguating English words were outperformed by a Lesk variant, used as baseline. On the other hand, at Senseval 2, Lesk baselines were outperformed by most of the systems in the lexical sample task.

The Lesk-based baselines outperform the baseline that uses simpler algorithms such as random sense assignment, or an algorithm that always chooses the sense which has most training-corpus instances.

The simplified Lesk (SL) algorithm (Kilgarriff and Rosenzweig, 1998) chooses the sense of an ambiguous word *w* such that its gloss *g* has the greatest number of words in common with other words (the context of *w*) around the given word *w*:

Table 3.1: Simplified Lesk algorithm

```
For each sense s of w do
  weight(s) = sim(c,g(s))
  s = argmax weight(s)
```

Here *c* is the context of the word *w* (in the simplest case, just a bag of words within a certain distance from *w*) and *g*(*s*) is the gloss associated with the sense *s*.

The Lesk-plus method (Kilgarriff and Rosenzweig, 1998) also considers a learning process, so it can be compared with supervised systems. For each word in the sentence containing the test item, it tests whether the word occurs in the dictionary entry or corpus instances for each candidate sense. For weighting of the sentences it uses the inverse document frequency (IDF) of a word, computed as log(*p*(*w*)), where *p*(*w*) is estimated as the fraction of dictionary "documents"—definitions or examples—which contain the word. Lesk-plus method does not explicitly represent the relative corpus frequencies of sense tags. Instead, it favours common tags because they have larger context sets, and an arbitrary word in a test-corpus sentence is more likely to occur in the context set of a more common training-corpus sense tag.

The original Lesk algorithm relies on glosses found in traditional dictionaries such as Oxford Advance Learner's dictionary. Banerjee and Pedersen (2002) propose a variant of the Lesk algorithm to take the advantage of the highly interconnected set of relations

among synonyms that WordNet offers. This variant takes as back-off the glosses of words that are related to the words to be disambiguated. This back-off provides a richer source of information and improves accuracy. It outperforms the baseline methods in the Senseval 2 exercise.

Vasilescu et al. (2004) proposed a set of different variants to the Lesk aproach. The first variant, the score assigned to a candidate sense is the number of overlaps between the BOW of that sense and the BOW of the context. A second variant, called WHG (for weighted) also takes into account the length of the description for a given sense. According to Lesk, long descriptions can produce more overlaps than short ones, and thus dominate the decision making process. Another type of variant multiplied the number of overlaps for a given candidate sense by the inverse of the logarithm of the description length for this sense. Other variant for weighting metrics were also proposed, taking into account the distance between a word in the context and the target word, or the frequency of the context word in the language, but that did not bring any significant difference.

In Statistical NLP, one commonly receives as a corpus a certain amount of data from a certain domain of interest, without having any say in how it is constructed. In such cases, having more training data is normally more useful than any concerns of balance, and one should simply use all the text that is available. The problem of data sparseness, which is common for much corpus-based work, is especially severe for work in WSD. First, enormous amounts of text are required to ensure that all senses of a polysemous word are represented, given the vast disparity in frequency among senses.

The Web is immense, free and available by mouse-click. It contains hundreds of billions of words of text and can be used for all manner of language research. The simplest language use is spell checking. Is it speculater or speculator? Google gives 67 for the first and 82,000 for the latter. Question answered.

3.2. Proposed Method

We augment the Lesk approach with a measure for semantic relatedness. The measure is based on the hypothesis of the high relationship between the gloss of a sense and the

context of a word. We measure this relationship by finding the frequencies of co-occurrences between the gloss and the context. We consider the Web as a corpus to find the co-occurrences of the gloss and the context.

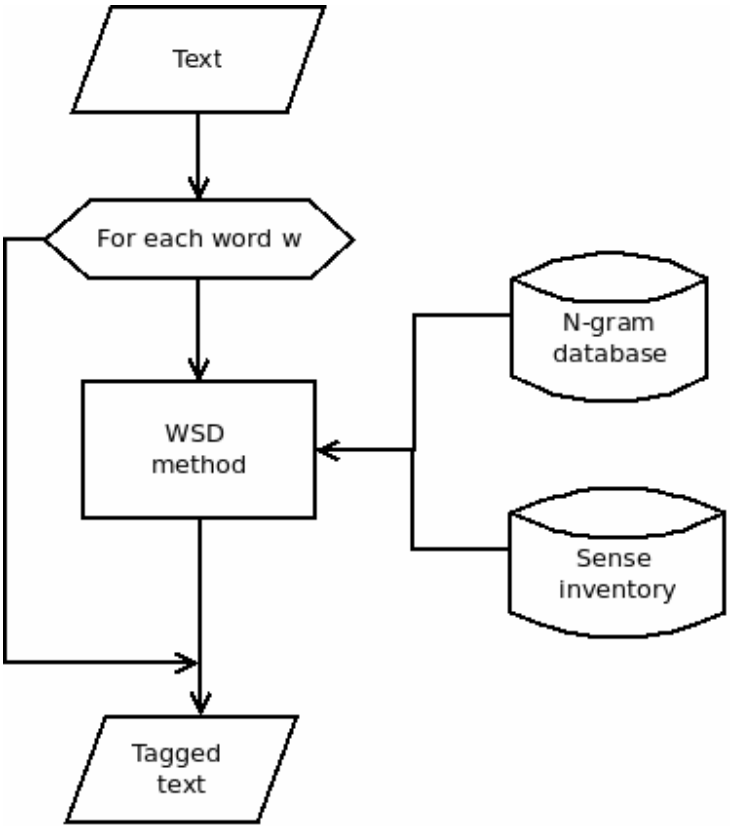

Figure 3.1: General data flow for the WSD method.

In Figure 3.1 we show the general scheme of our proposed method. For each word in the text the method tagged the word with a sense from a sense inventory. The method takes decisions based on: the n-gram database and the sense inventory. We used for n-gram database the Web and for sense inventory WordNet. Below we show how the new measure can be applied to our method (Simple Lesk algorithm):

Table 3.2: New statistical measure

```
For each word w to be tagged
 For each sense s of w
    g = gloss of sense s (bag of words)
    e = example of sense s (bag of words)
    c = context of sense s (bag of words)
    d = g ∪ e
    dc = d ∪ c
    f_g  = web frequency of(d)
    f_gc = web frequency of(dc)
    weight(s) = f_gc/f_g
 S  = arg max weight (s)
```

The web frequency is measured by a query to a web search engine. The weight is the probability of seeing the gloss of a sense in the context of the given word occurrence. The method chooses the sense which maximizes the weight.

If various senses have the same weight, then the sense is chosen by a back-off heuristic.

3.3. Experimental Setting

In this subsection firstly we show a brief description of the datasets used, second the experimental setting of the proposed measure and finally a comparison with previous results.

3.3.1. Data Sets

Semcor is a textual corpus in which words are syntactically and semantically tagged. The texts included in Semcor were extracted from the Brown corpus and then linked to senses in the WordNet lexicon. All the words in the corpus have been syntactically tagged using Brill's part of speech tagger; the semantically tagging was done manually for all the nouns,

verbs, adjectives and adverbs, each of these words being associated with its correspondent WordNet sense. We show above an example of an entry in the Semcor corpus.

<wf cmd=done pos=VB lemma=say wnsn=1 lexsn=2:32:00::>said</wf>

The Senseval dataset consists of 4,328 instances each of which contains a sentence with a single target word to be disambiguated, and one or two surrounding sentences that provide additional context.

A task in Senseval consists of three types of data: 1) A sense inventory of word-to-sense mappings, with possibly extra information to explain, define, or distinguish the senses (e.g., WordNet); 2) A corpus of manually tagged text or samples of text that acts as the Gold Standard, and that is split into an optional training corpus and test corpus; and 3) An optional sense hierarchy or sense grouping to allow for fine or coarse grained sense distinctions to be used in scoring. The next XML is an example of an entry in Senseval.

<instance id="9:0@16@wsj/24/wsj_2444@wsj@en@on" docsrc="wsj">

<context>

Once metropolitan ...<head> asking </head> ...

</context>

</instance>

Senseval has two variants of the WSD task:

All words task participating systems have to disambiguate all words (open-class words) in a set of text, and Lexical sample task, first a sample of words is selected. Then for each sample word, a number of corpus instances are selected.

3.4. Experimental Results

In our preliminary experiments we aimed at the all words WSD task. For evaluation we used a subset of the first two tagged files of SemCor 1.6: the files br-a01 and br-a02. We used WordNet 2.1 as a sense repository. WordNet is a lexical database where each unique meaning of a word is represented by a synonym set or synset. Each synset has a gloss that defines the concept that it represents. For example, the words *car, auto, automobile*, and

motorcar constitute a single synset that has the following gloss: *four wheel motor vehicle, usually propelled by an internal combustion engine*. Many glosses have examples of usages associated with them, such as "*he needs a car to get to work*."

Context is the only means to identify the meaning of a polysemous word. Therefore, all work on WSD relies on the context of the target word to provide information to be used for its disambiguation. Most disambiguation work uses the local context of a word occurrence as a primary information source for WSD. Local or "micro" context is generally considered to be some small window of words surrounding a word occurrence in a text or discourse, from a few words of context to the entire sentence in which the target word appears.

Context is very often regarded as all words or characters falling within some window of the target, with no regard for distance, syntactic, or other relations. Yarowsky (1993) examines different windows of micro-context, including 1-contexts, k-contexts, and words pairs at offsets -1 and -2, -1 and + 1, and +1 and +2, and sorts them using a log-likelihood ratio to find the most reliable evidence for disambiguation. Yarowsky makes the observation that the optimal value of k varies with the kind of ambiguity: he suggests that local ambiguities need only a window of k = 3 or k = 4.We use the bag of words approach: here, context is considered as words in some window surrounding the target word, regarded as a group without consideration for their relationships to the target in terms of distance, grammatical relations. We take a symmetric window of ±3 words around the target word an optimal value to local ambiguities.

The web counts were collected using the Google1 search engine. To construct the queries first we tokenize the sentence, second the target word is replaced by the gloss, and query the search engine (i.e. string query).

When our method can not decide a sense number in the argmax function (e.g. two senses have the same weight), the sense will be chosen randomly from the top senses (those with the same weight); we refer to this as random top weight back-off.

We used precision and recall to score the system, although the metrics are not completely analogous to Information Retrieval evaluation. Recall (percentage of right answers on all instances in the test set) is the basic measurement of accuracy in this task, because it shows how many correct disambiguations the system achieved overall. Precision

[1] http://www.google.com

(percentage of right answers in the set of answered instances) favours systems that are very accurate if only on a small subset of cases that the system chose to give answers to.

Resnik and Yarowsky (1993) have shown that it is difficult to compare WSD methods. The distinctions that make comparing methods difficult reside in the approach considered (supervised or unsupervised).

The result from the preliminary experiments over the SemCor subset obtained an accuracy of 47%. We only reported accuracy because of any word presented an equal weight of senses. If a system makes an assignment for every word, then precision and recall are the same, and can be called accuracy. Therefore the Web rarely presents data sparseness. Thus the method always gives an answer and it does not reach the back-off heuristic.

In Table 3.3 we present a comparison of the accuracy of our measure applied to the simple Lesk against variants of the original Lesk approach. This comparison was tested over the Senseval 2 data. The experiment had the same setting as the experiment over the SemCor subset.

Table 3.3: Comparison with previous work

Method	Type	Back-off	Accuracy
Vasilescu et al. 2004	simplified	MFS	0.58
Mihalcea and Tarau 2004	simplified	RS	0.47
Our method	simplified	Random top senses	0.45
Vasilescu et al. 2004	original	MFS	0.42
Mihalcea and Tarau 2004	original	RS	0.35
Banerjee and Pedersen 2002	original	Extended gloss overlaps	0.31

As it can bee seen from Table 3.3, the original Lesk (OL) algorithm method has a lower performance than the other ones and even than the baseline system. This observation is

consistent with Litkowski (2002) hypothesis that only about one third of the instances can rely on the Lesk-style information (gloss and example) in a disambiguation process. The simplified Lesk (SL) method, which only counts the overlaps between the description of a candidate sense and the words in the context, produces better results.

Our Lesk variant outperforms the OL of Banerjee and Pedersen (2002) and the OL (Lesk, 1986) variants (back-off to random sense and most frequent sense). The SL of Mihalcea and Tarau (2004) is better in performance than our method, with the help of the random sense heuristic. Finally, the SL of Vasilescu et al. (2004) has the best accuracy. However, this method can be considered as a supervised method due to the most frequent sense heuristic (it is not clear what its performance would be with McCarthy et al. (2004) unsupervised method for determining the predominant sense).

When a method can not make a judgment (i.e., no overlap between the gloss and the context in the simple Lesk) the judge is taken by the back-off heuristic. Most of these heuristics chose a random sense or uses information from a dictionary. So the most frequent sense is based on chose the first (or predominant) sense the heuristic assumes the availability of hand tagged data.

Therefore our method did not reach the back-off heuristic we present in Table 3.4 a comparison with the top three unsupervised methods of Senseval-2.

Table 3.4: Comparison with Senseval-2 unsupervised methods

Method	Accuracy
Our method	0.45
Senseval–First	0.40
Senseval–Second	0.29
Senseval–Third	0.24
Original Lesk	0.18

The Senseval–First, Senseval–Second, and Senseval–Third results are the top three most accurate fully automatic unsupervised systems in the Senseval-2 exercise. This is the class of systems could be comparable to our own, since they require no human intervention

and do not use any manually created training examples. These results show that our approach was considerably more accurate than all. This method has the advantage of simplicity and the use of a very limited context window.

Table 3.5: Comparison with SemEval unsupervised methods

Method	Accuracy
Radu ION	0.52
Davide Buscaldi	0.46
Our method (Senseval-2)	0.45
Sudip Kumar Naskar	0.40

In Table 3.5 we present a comparison of our method (tested over Senseval-2) with the state of the art unsupervised systems in the SemEval-2007. Thus two of the methods outperform our method but the comparison is not so clear because our method was test over the Senseval-2 corpus. Thus we can see growing tendencies in the precision of the unsupervised approaches.

3.5. Conclusion

In this thesis we used a variant of the Lesk algorithm for the WSD task. We proposed a new semantic relatedness measure based on web counts collected with a search engine.

We have shown that our variant outperforms some Lesk based methods and outperforms the top unsupervised methods of the Senseval-2 exercise. These results are significant because they are based on a very simple algorithm that relies on co-occurrences scores to the senses of a target word.

We once more confirmed that the web could be used as a lexical resource for WSD.

In our future work we will explore the use of different context windows, as well as linguistically-motivated context windows (such as a syntactic unit) and test our method over the SemEval corpus.

4. Recognizing Textual Entailment

In this chapter we proposed a new statistical method applied to the RTE task. The new statistical method is based on the co-occurrences of words between the T-H pairs. The co-occurrences are extracted from a cause-effect corpus. Therefore we are deciding the entailment based on a non-symmetric measure of similarity. Follow as first we show the experiments over the RTE-1, RTE-2, and RTE-3 test datasets and as a second experiment we proposed a meta-classifier, based on symmetric and non-symmetric measures of similarity. Finally we compare our methods with previous works and draw partial conclusions.

4.1. Theoretical Framework

As we could see in Chapter 2 the main methods for RTE are based on the level of representation given to the T-H pair. Thus each type of representation has operations in order to establish the entailment decision (e.g., word matching in the lexical level, tree edit distance in the syntactic level). The principal operations are similarity measures between T-H pair representations. But many of the similarity measures are symmetric. So a symmetric measure can not capture some of the aspects in the T → H relation. Because if we altered the entailment relation (i.e., H → T) a symmetric function will give us the same score. Therefore methods like (Tatar et al. 2007) propose a non-symmetric similarity measure, used in RTE-1 Challenge.

Glickman uses as definition: T entails H iff P(H | T) > P(H). The probabilities are calculated on the base of Web. The accuracy of the system is the best for RTE-1 (0.56).

Another non-symmetric method is that of Kouylekov, who uses the definition: T entails H iff there exists a sequence of transformations applied to T such that H is obtained with a total cost below of a certain threshold. The following transformations are allowed: Insertion: insert a node from the dependency tree of H into the dependency tree of T; Deletion: delete a node from the dependency tree of T; Substitution: change a node in the T into a node of H. Each transformation has a cost and the cost of edit distance between T and H, ed(*T*, *H*) is the sum of costs of all applied transformations. The entailment score of a given pair is calculated as

$$\text{score}(T,H) = \text{ed}(T,H),$$

where ed(· , *H*) is the cost of inserting the entire tree H. If this score is bigger than a learned threshold, the relation T → H holds. The accuracy of method is of 0.56.

In (Corley and Mihalcea, 2005) an even "more non-symmetric" is proposed: when the edit distance (which is a Levenshtein modified distance) satisfies the relation:

$$\text{ed}(T,H) < \text{ed}(H,T),$$

then the relation T→ H holds.

Other authors use a definition which in terms of representation of knowledge as feature structures could be formulated as: T entails H iff H subsumes T. Even the method used is a non-symmetric one, as the definition used is: T entails H iff H is not informative in respect to T.

A method of establishing the entailment relation could be obtained using a non-symmetric measure of similarity between two texts presented by Corley and Mihalcea (2005), the authors define the similarity between the texts T_i and T_j with respect to Ti as:

$$sim(T_i, T_j)_{Ti} = \frac{\sum_{pos}\left(\sum_{wk \in ws_{pos}^{ti}} (\max Sim(w_k) \times idf(w_k))\right)}{\sum_{pos}\sum_{wk \in ws_{pos}^{ti}} idf(w_k)}$$

Here the sets of open-class words (nouns, verbs, adjective and adverbs) in each text segment. For a word wk with a given pos in T_i, the highest similarity of the words with the same pos in the other text T_j is denoted by maxSim(w_k).

Starting with this text-to-text similarity metric, we derive a textual entailment recognition system by applying the lexical refutation theory presented above. As the

hypothesis H is less informative than the text T, for a TRUE pair the following relation will take place:

$$\text{sim}(T,H) \times T < \text{sim}(T,H) \times H$$

This relation can be proven using the lexical refutation. A draft is the following: to prove T → H it is necessary to prove that the set of formulas {T; negH} is lexical contradictory (they denote also by T and negH the sets of disjunctive clauses of T and negH).

We propose a new non-symmetric measure of similarity based on the co-occurrences of words between the T-H pair in a cause-effect corpus. Follow the use of the two types of similarity measures (symmetric, non-symmetric) in a meta-classifier.

4.2. Proposed Methods

Before the presentation of the new methods we show a brief introduction to the main evaluation measures. This evaluation measures are used to evaluate the performance of the proposed methods.

An important recent development in NLP has been the use of much more rigorous standards for the evaluation of systems. It is generally agreed that the ultimate demonstration of success is showing improved performance at an application task, be that spelling correction, summarizing job advertisements, or whatever. Nevertheless, while developing systems, it is often convenient to assess components of the system on some artificial performance score (such as perplexity), improvements in which one can expect to be reflected in better performance for the whole system on an application task.

Evaluation in IR makes frequent use of the notions of precision and recall, and their use has crossed over into work on evaluating Statistical NLP models, such as a number of the methods discussed in this chapter. For many problems, we have a set of targets (for example, targeted relevant documents, or sentences in which a word has a certain sense) contained within a larger collection. The system then decides on a selected set (documents that it thinks are relevant, or sentences that it thinks contain a certain sense of a word, etc.).

The selected and target groupings can be thought. The variables can be expressed as a 2x2 contingency matrix:

Table 4.1: Contingency matrix

	Corpus	
System	true	false
true	tp	fp
false	fn	tn

The numbers in each box show the frequency or count of the number of items in each region of the space. The cases accounted for by *tp* (true positives) and *tn* (true negatives) are the cases our system got right. The wrongly selected cases in *fp* are called false positives, acceptances or Type errors. The cases in *fn* that failed to be selected are called false negatives, false rejections or Type I errors.

The accuracy is the proportion of true results (both *tp* and *tn*) in the population:

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn}$$

Precision is defined as a measure of the proportion of selected items that the system got right:

$$precision = \frac{tp}{tp + fp}$$

Recall is defined as the proportion of the target items that the system selected:

$$recall = \frac{tp}{tp + fn}$$

It can be convenient to combine precision and recall into a single measure of overall performance. One way to do this is the F-measure. The F-measure is defined as follows (precision and recall have and equal weighting):

$$F = \frac{2PR}{R + P}$$

4.2.1. Causal Non-symmetric Measure

A causal relation is the relation existing between two events such that one event causes (or enables) the other event, such as “hard rain causes flooding” or “taking a train requires buying a ticket”. The idea behind knowledge acquisition is to use connective markers such as “because”, “but” and “if” as linguistic cues. However, there is no guarantee that a given connective marker always signals the same type of causal relation. In this thesis we focused our attention on English sentences including the word “because”.

Consider the following examples in English, from which one can obtain several observations about the potential sources of causal knowledge.

- *The laundry dried well today because it was sunny.*
- *The laundry dried well, though it was not sunny.*
- *If it was sunny, the laundry could dry well.*
- *The laundry dried well because of the sunny weather.→ Cause(it is sunny , laundry dries well).*

Cue phrase is a word, a phrase or a word pattern, which connects one event to the other with some relation. The causal relation between events is assumed by the cue phrase. The causal cue phrases are used for connecting the cause and effect events. When events are expressed bye noun phrases, the cue phrase connecting events is a verb phrase in general. For example:

- *The oral bacteria that cause gum disease appear to be the culprit.*

The verb “cause” is a cue phrase to connect two events expressed by noun phrases, the “oral bacteria” and “gum disease”. Several lexical pairs are assumed to lead the causal relation. The lexical pair “bacteria” and “disease” is an example of the causal lexical pair. If the term pair “the oral bacteria” and “gum disease” is causally related, we can infer the event pair “bowel bacteria” and “bowel disease” is causally related. Causal lexical pairs are learned from cause-effect pairs.

The causal relation subsumes the cause and the explanation relations in Hobbs (1985). Hobbs’s cause relation holds if a discourse segment stating a cause occurs before a discourse segment stating an effect; an explanation relation holds if a discourse segment

stating an effect occurs before a discourse segment stating a cause. The causal relation is encoded by adding a direction. In a graph, this can be represented by a directed arc going from cause to effect, Fig 4.1.

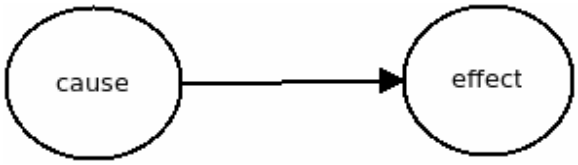

Figure 4.1: Cause effect graph

The hypothesis behind our method is based on treat the T-H pair as a causal relation. Where the text T is a cause and the hypothesis H is its effect (i.e., T causes H).

The non-symmetric similarity measure is based on the count of co-occurrences of causal lexical pairs from a C-E pairs extracted from a corpus.

Table 4.2: Non-symmetric similarity measure

For each word t_i in T
  For each word h_j in H
  ce_j = causal frequency(t_i,h_j)
  e_j = causal frequency(h_j)
  max_i = argmax(ce_j/e_j)
non-symmetric(T,H)= Σ max_i

As we see in the table 4.2 the first causal frequency function is the count of words t_i and h_i related by the cue phrase (For example, a sentence, h…because…t) in a corpus of C-E pairs and the second causal frequency function is the count of word h_i in the C-E pairs, which gives us a non-symmetric score. Because the co-occurrences of T causes H is not the same like H causes T.

To each T-H pair the system measures the causal relation between them and then decides if the pair is true or false given a certain entailment decision. The main differences in our experiments reside in the use of different strategies to decide the entailment relation.

4.2.2. Experimental Setting

In this subsection we explain at detail some of the blocks in the Figure 4.2. First the pre-processing we used to represent the T-H pair and second the data used to create the C-E pairs.

The prepossessing we used in each T-H pair to be tagged is as follows:

- Tokenize.
- Quit stop words.

Normally, an early step of processing is to divide the input text into units called tokens where each is either a *word* or something else like a number or a punctuation mark. This process is referred to as the treatment of punctuation varies.

The system has just stripped the punctuation out. We consider as word any object within the occurrence of a withespace. The withespace is the main clue used in English (RTE benchmark is in English). Finally the system quits any stops words from a stoplist. The stop word is the name given to words which are filtered out prior to, or after, processing of text. Hans Peter Luhn, one of the pioneers in information retrieval, is credited with coining the phrase and using the concept in his design. It is controlled by human input and not automated (Manning and Shutze, 1999). Common stop words are *the*, *from* and *could*. These words have important semantic functions in English, but they rarely contribute information if the criterion is a simple word-by-word match (Manning and Shutze, 1999). In Figure 4.2 we show the general data flow of our method were the data used to collect the frequency of the causal lexical pairs came from sentences which contain the cue phrase *because*.

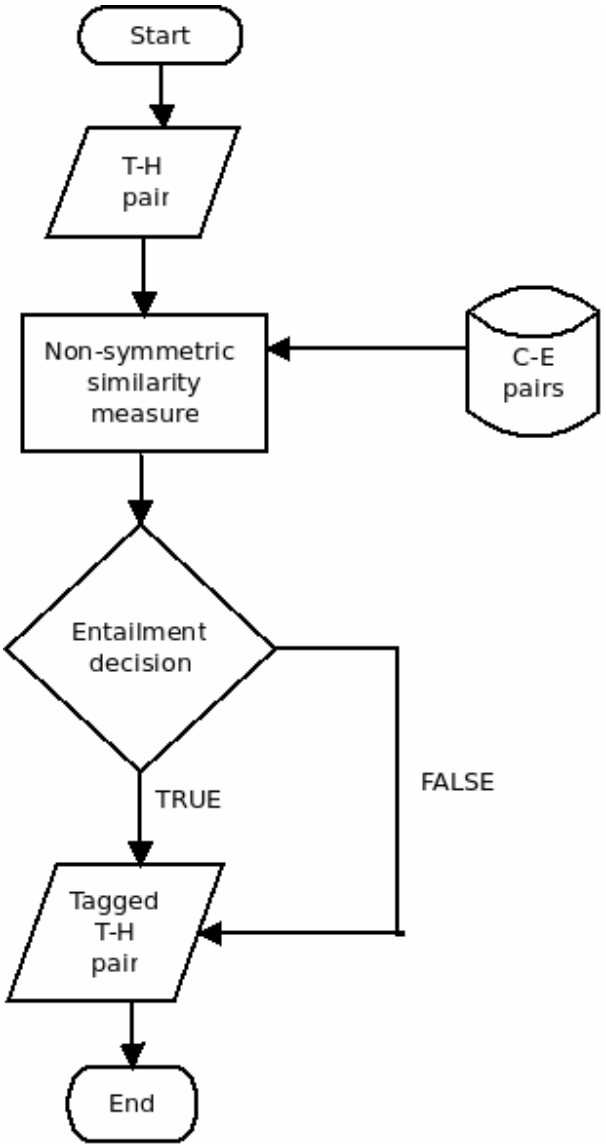

Figure 4.2: General data flow of our system

The sentences were extracted from the Sketch Engine system from a big corpus (ukWAC from the Sketch Engine http://www.sketchengine.co.uk/) for sentences which contains the discourse marker *because*. Finally we striped the sentences in two parts: one corresponding to the cause and one corresponding to its effect. The Sketch Engine is a corpus query system which allows the user to view word sketches, thesaurally similar words, and 'sketch differences', as well as the more familiar Corpus Query Systems (CQS) functions. The word sketches are fully integrated with the concordancing: by clicking on a collocate of interest in the word sketch, the user is taken to a concordance of the corpus evidence giving rise to that

collocate in that grammatical relation. If the user clicks on the word *toast* in the list of high-salience objects in the sketch for the verb *spread*, they will be taken to a concordance of contexts where *toast (n)* occurs as object of *spread (v).*

4.2.3. Experimental Results

As we see in previous subsections we varied the entailment decision in order to prove some differences between the uses of our non-symmetric measure:

- Experiment 1: The system penalizes a pair if the H → T relation is greater than T → H relation.
- Experiment 2: The system determines the entailment decision based on a certain threshold (learned from corpus).

The outline of the information displayed on each experiment is the next one:

- Contingency matrix.
- Evaluation matrix.
- Comparison with previous wok.
- Accuracy depending on task.

The experiments are divided by the RTE Challenge versions (i.e. RTE-1, RTE-2, and RTE-3).

4.2.3.1. Experiment 1

The first experiment is based on use the non-symmetric measure. The entailment decision is as follows:

Table 4.3: Entailment decision 1

if non-symmetric(*T*,*H*) > non-symmetric(*H*,*T*) then TRUE else FALSE

In table 4.3 we see that the entailment decision is basically penalize a T—H pair when the H → T relation is stronger than the T → H relation. Therefore the hypothesis H is more probably an effect than the text T. Therefore it is more probable that the text T implies the hypothesis H. First, we present the method applied to the RTE-1. The contingency table, Table 4.4 show how many times the method misclassified the T-H pairs (i.e. *fp* and *tn*) and how many times the method its right. From this table we can obtain some measures to evaluate the entailment decision.

Table 4.4: RTE-1 contingency matrix results

	true	false
true	257	245
false	143	155

Table 4.4 also shows that our approach tends to say true.

Table 4.5: RTE-1 evaluation measures

Accuracy	Precision	Recall	F-measure
0.51	0.51	0.64	0.57

From table 4.5 this approach obtains a better recall than precision. Therefore the entailment decision got right the proportion of the target items that the system selected.

Table 4.6: RTE-1 comparison with previous results

Method	Accuracy
GLICKMAN	0.56
LEVENSHTEIN	0.53
C-E	0.51
BLEU	0.49

To compare our approach with previous works we use the accuracy measure (i.e. the most common measure in the RTE Challenge).The proposed measure is compared to non-symmetric measure. We compare out approach with

- Bleu algorithm RTE baseline (Perez and Alfonseca. 2005).
- Probabilistic measure (Glickman et al 2005).
- Levenshthein modified measure (Tatar et al. 2007).

In table 4.6 the results are show. Thus the best one is Glickman. Our measure is the last one compare to the non-symmetric measures. Our measure only outperforms the Bleu algorithm.

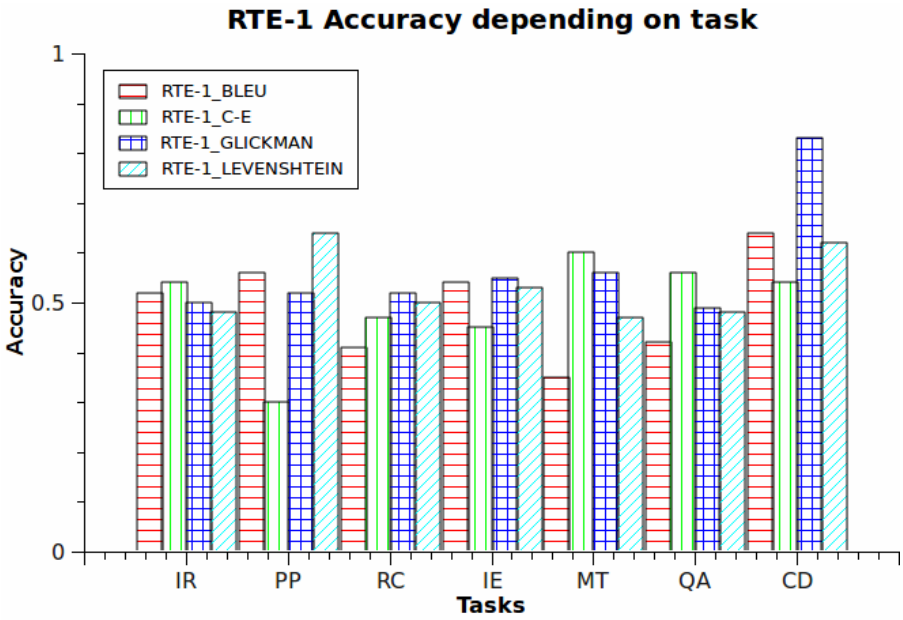

Figure 4.3: RTE-1 comparison with previous results by tasks

The results of our approach were the lowest between the non-symmetric measures in general. So if we make a comparison depending on each task. We see that our measure outperforms the other non-symmetric measures in some of the tasks. These tasks are:

- QA.
- IR.
- MT.

For the RTE-2 the scores did not varied too much. The tendency to say true of the method is the same, Table 4.7.

Table 4.7: RTE-2 contingency matrix

	true	false
true	289	271
false	111	129

Table 4.8: RTE-2 evaluation measures

Accuracy	Precision	Recall	F-measure
0.52	0.51	0.72	0.60

The recall and precision increases in comparison with the past data set, Table 4.8.

Table 4.9: RTE-2 comparison with previous results

Method	Accuracy
CORLEY	0.58
BLEU	0.53
C-E	0.52

For the RTE-2 we compare our method with:

- Bleu algorithm.
- WordNet modified similarity measures (Corley and Mihalcea, 2005).

Also out method only outperforms the baseline.

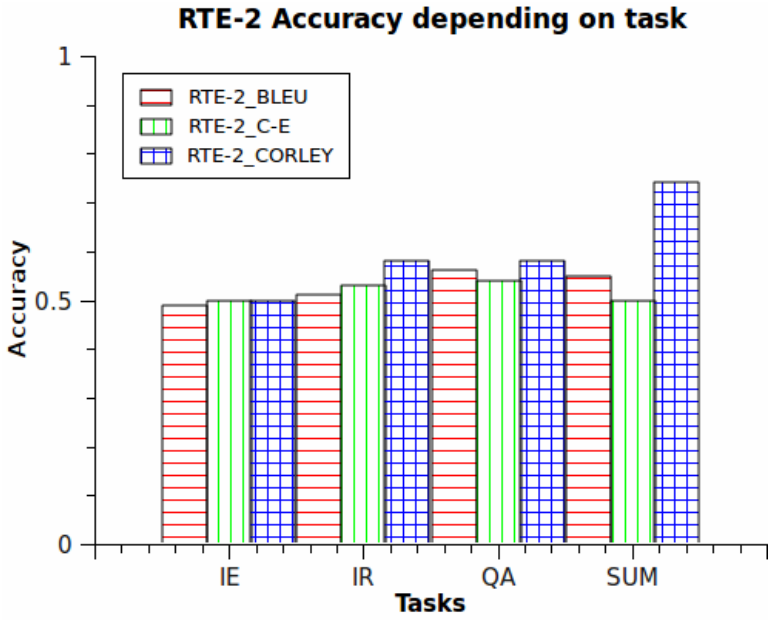

Figure 4.4: RTE-2 comparison with previous results by tasks

In the comparison by task our method did not pout perform the method of Corley et al. (2009).

InRTE-3 the tendency to say true is the same. The tendency to say true increase in every data set.

Table 4.10: RTE-3 contingency matrix

	true	false
true	303	268
false	107	122

Table 4. 11: RTE-3 evaluation measures

Accuracy	Precision	Recall	F-measure
0.53	0.53	0.73	0.61

Also the results did not varied too much from other evaluations.

Table 4.12: RTE-3 comparison with previous results

Method	Accuracy
CORLEY	0.63
C-E	0.53
BLEU	0.50

Corley's measure is still the best one. The measure reaches the highest accuracy over the tree data sets.

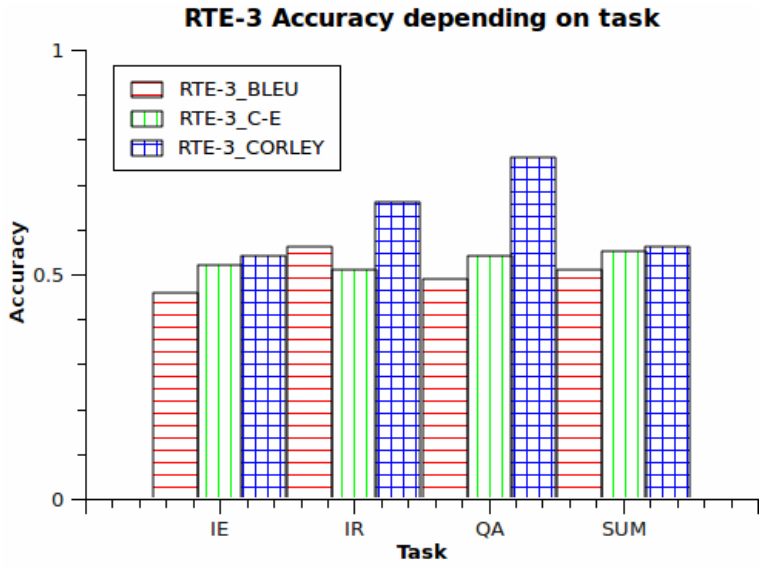

Figure 4.5: RTE-3 comparison with previous results by task

Almost the same result than the past data set we can not outperform the modified WordNet measure from Corley et al. (2009).

4.2.3.2. Experiment 2

The second experiment is based on use the non-symmetric measure. The entailment decision is as follows:

Table 4.13: entailment decision 2

if non-symmetric(*T*,*H*) > threshold then
TRUE
else
FALSE

In table 4.13 we see that the entailment decision is similar to a symmetric approach. We take a threshold from the test set to decide if a T-H pair is true or false. First we introduce the results of using different thresholds:

Table 4.14: RTE evaluation measures with entailment decision 2 and a threshold of 0.1

Dataset	Accuracy	Precision	Recall	F-measure
RTE-1	0.51	0.51	0.80	0.62
RTE-2	0.50	0.50	0.75	0.60
RTE-3	0.53	0.53	0.7	0.63

Table 4.15: RTE evaluation measures with entailment decision 2 and a threshold of 0.2

Dataset	Accuracy	Precision	Recall	F-measure
RTE-1	0.50	0.50	0.56	0.53
RTE-2	0.50	0.50	0.50	0.50
RTE-3	0.53	0.54	0.53	0.53

Table 4.16: RTE evaluation measures with entailment decision 2 and a threshold of 0.3

Dataset	Accuracy	Precision	Recall	F-measure
RTE-1	0.51	0.52	0.40	0.45
RTE-2	0.50	0.51	0.33	0.40
RTE-3	0.52	**0.55**	0.36	0.43

The decision of the threshold used on the comparison experiments was to maximize the precision. Because in previous experiments the precision measure was very low.

Table 4.17: RTE-1 contingency matrix

	true	false
true	161	147
false	239	253

In Table 4.17 we can see that the tendency change from true to false. Thus this approach is stricter with tag a T-H pair as “true”.

Table 4.18: RTE-1 comparison with previous results

Method	Accuracy
GLICKMAN	0.56
LEVENSHTEIN	0.53
C-E	0.51
BLEU	0.49

Comparing to the other measures there is no significant changes in accuracy.

In the comparison depending on task our method outperforms in:

- IE.
- QA.

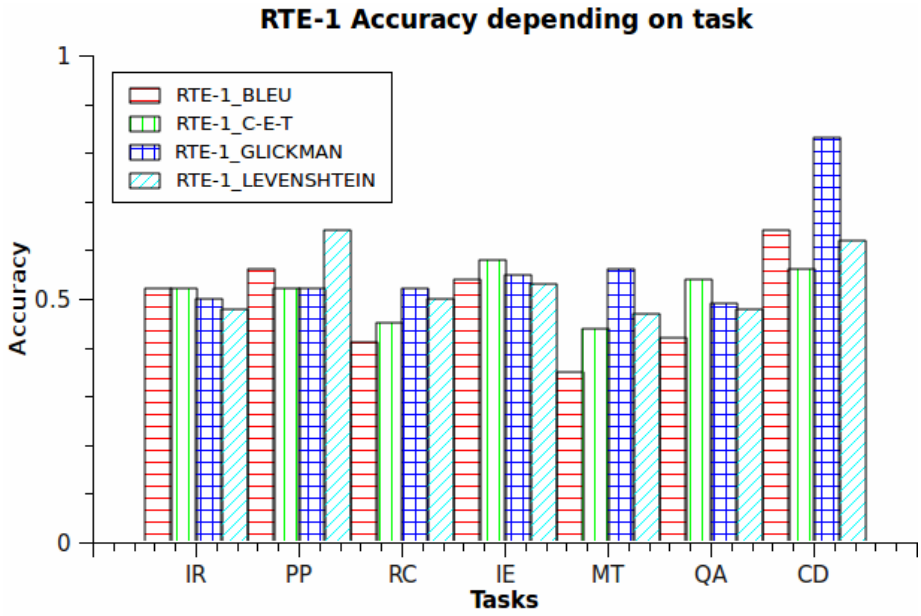

Figure 4.6: RTE-1 comparison with previous results by tasks

For the RTE-2 the tendency to say false continues, Table 4.19.

Table 4.19: RTE-2 contingency table

	true	false
true	161	147
false	239	253

Table 4.20: RTE-2 evaluation with previous results

Method	Accuracy
GLICKMAN	0.56
LEVENSHTEIN	0.53
C-E	0.50
BLEU	0.49

Out method outperforms the bleu algorithm but is the lower to the other non-symmetric measures. The threshold approach which is more similar to the other non-symmetric approaches did not show to be good in this data set.

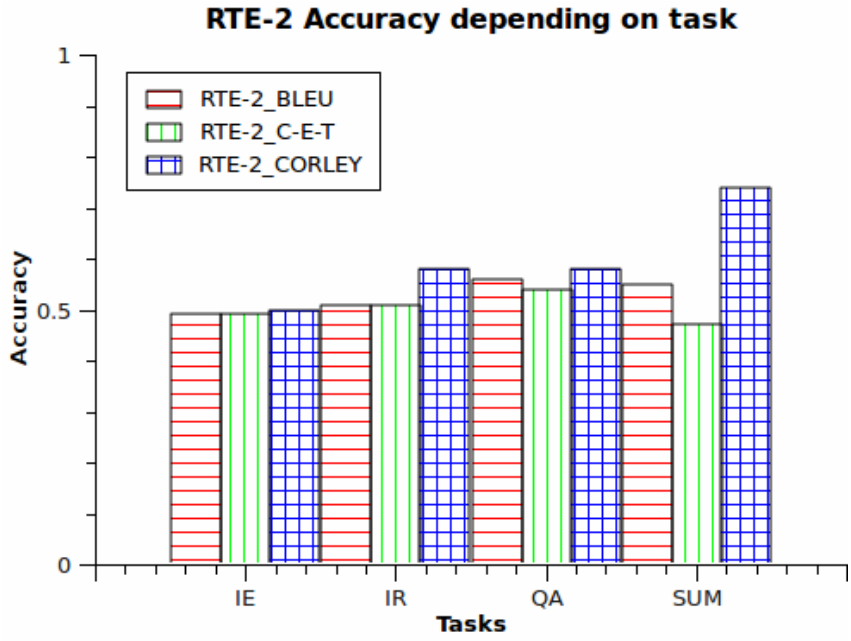

Figure 4.7: RTE-2 comparison with previous results by tasks

In Figure 4.7 like the other datasets our method did not outperform in any task.

For the RTE-3 the results did not varied respect to the other data sets, Tables 4.21, 4.22, and Figure 4.8.

Table 4.21: RTE-3 contingency matrix

	true	false
true	161	147
false	239	253

Table 4.22: RTE-3 comparison with previous results

Method	Accuracy
GLICKMAN	0.56
LEVENSHTEIN	0.53
C-E	0.52
BLEU	0.49

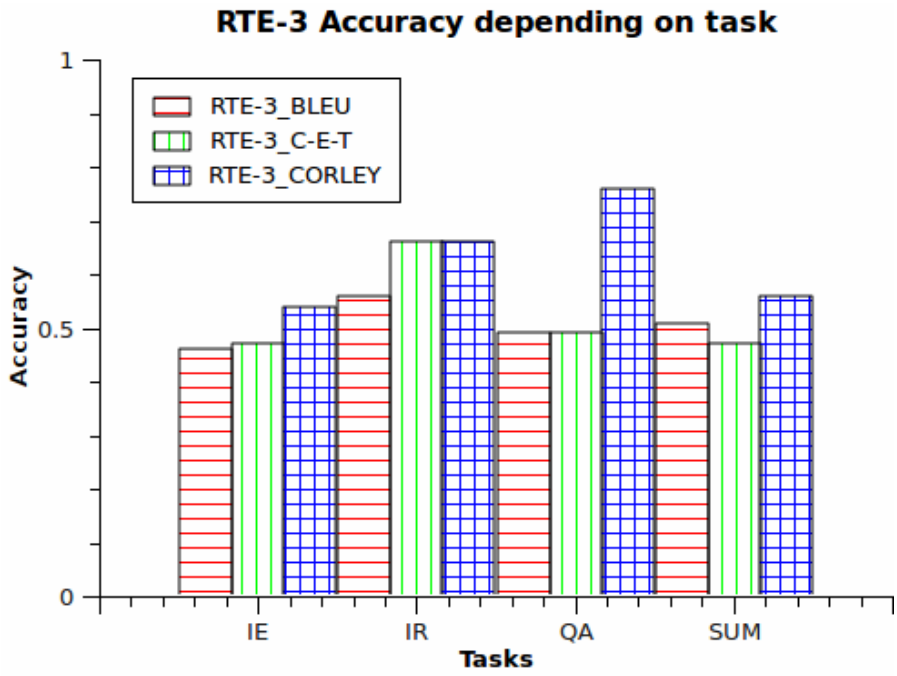

Figure 4.8: RTE-3 comparison with previous results by tasks

Therefore these results show us the low performance of our measure. We decide to make some other experiments now over the joint use of the symmetric and non symmetric measures.

4.2.4. Symmetric and Non-symmetric Meta-classifier

It has been observed for related systems that a combination of separately trained features in the machine learning component can lead to an overall improvement in system performance, in particular if features from a more informed component and shallow ones are combined (Hickl et al. 2006; Bos and Markert, 2006 and Butchart 2007).

One of the main problems when machine-learning classifiers are employed in practice is to determine whether classifications assigned to new instances are reliable. The meta-classifier approach is one of the simplest approaches to this problem. Given a base classifiers, the approach is to learn a meta-classifier that predicts the correctness of each instance classification of the base classifiers. The sources of the meta-training data are the training instances. The meta-label of an instance indicates reliable classification, if the

instance is classified correctly by a base classifier; otherwise, the meta-label indicates unreliable classification. The meta-classifier plus the base classifiers form one combined classifier. The classification rule of the combined classifier is to assign a class predicted by the base classifier to an instance if the meta-classifier decides that the classification is reliable.

Thus some questions on how to design a meta-classifier are:

- What type of base classifiers do we have to learn for meta-classifier, for what type of data?
- What is the role of the accuracy of the base classifiers in the whole scheme?
- How do we have to represent meta-data?
- How can we have to generate meta-data?

4.2.5. Experimental Design

To answer the previous questions we designed a meta-classifier as follows:

- We used symmetric and non-symmetric measures as base classifiers.
- We chose the best symmetric measure (optimizing accuracy).
- We represented the T-H pairs as a BoW.
- We used as meta-data the RTE Challenge test sets.

4.2.6. Experimental Results

In this subsection we compare the results of the meta-classifier to each base classifier. Also like in the previous part we develop two experiments:

- Experiment 1: non-symmetric measure without threshold (base classifier) and symmetric cosine.
- Experiment 2: non-symmetric measure with threshold (base classifier) and symmetric cosine.

4.2.6.1. Experiment 1

The first experiment uses as base classifiers the cosine measure and our non-symmetric measure (Experiment 1: T → H > H → T).

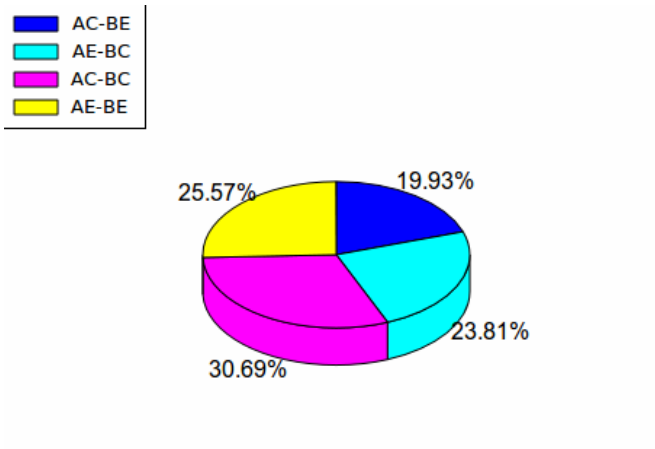

Figure 4.9: RTE-1 meta-classifier coverage

In figure 4.9 it is show percentage of the coverage of the different base classifiers over the RTE-1 development data set. Where *A* means our approach and *B* means the cosine measure. Thus *C* means correct classification and *E* means a misclassification. For example, AC means that the non-symmetric measure got a right classification.

Therefore more T-H pairs could be resolved also by the symmetric and the non-symmetric measures. Following the examples resolved by the symmetric measure and the non-symmetric at last. Finally the 25.57% of the instances could not be resolved by any measure.

Table 4.23: RTE-1 contingency table

	true	false
true	279	247
false	121	153

So our meta-classifier has the tendency to say true, Table 4.23 Thus from table 4.24 we see an overall increase in the evaluation measures over the test data set.

Table 4.24: RTE-1 evaluation measures

Accuracy	Precision	Recall	F-measure
0.54	0.53	0.68	0.60

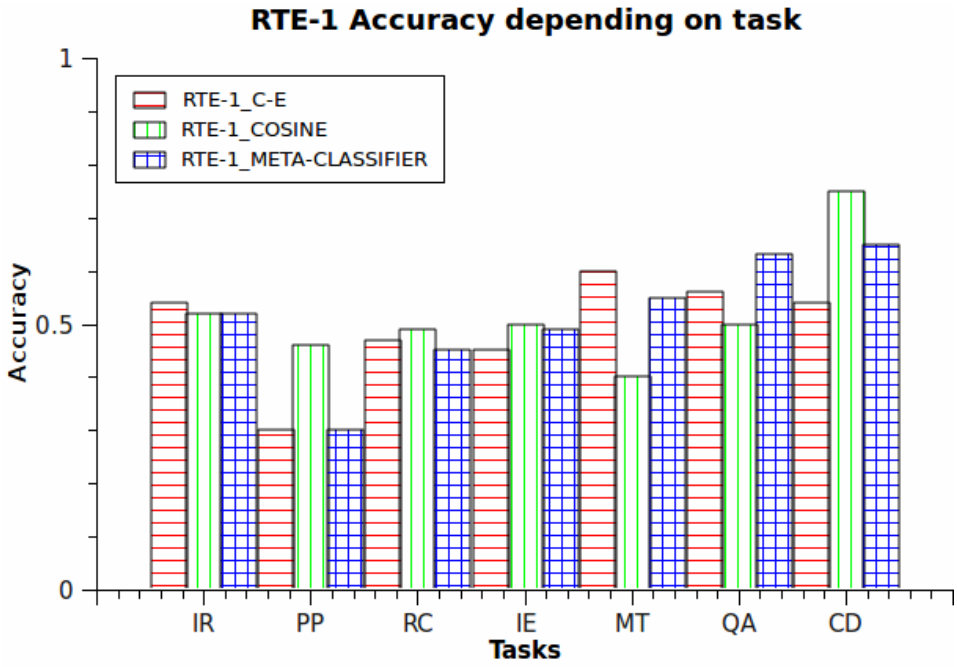

Figure 4.10: RTE-1 comparison with previous results

The meta-classifier only outperform in the QA task.

The cosine measure by itself outperform in the following tasks:

- PP.
- RC.
- IE.
- CD.

The non-symmetric measure outperform in the following tasks:

- IR.
- MT.
- QA.

In the RTE-2 development data set the coverage values are the next: Figure 4.11.

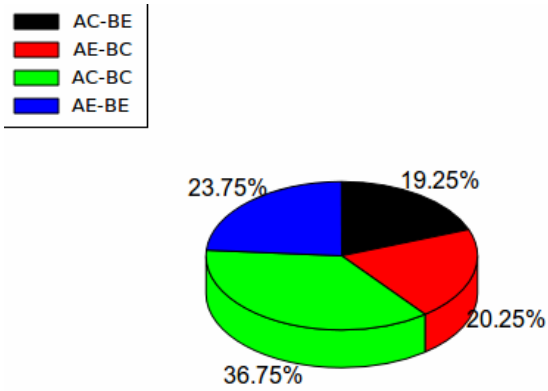

Figure 4.11: RTE-2 meta-classifier coverage

The behaviour of the coverage is similar to the RTE-1 development data set. Where most of the T-H pairs can be resolved by the two paradigms: the symmetric measure and the non-symmetric measure.

Table 4.25: RTE-2 contingency matrix

	true	false
true	296	267
false	104	133

Table 4.26: RTE-2 evaluation measures

Accuracy	Precision	Recall	F-measure
0.53	0.52	0.70	0.61

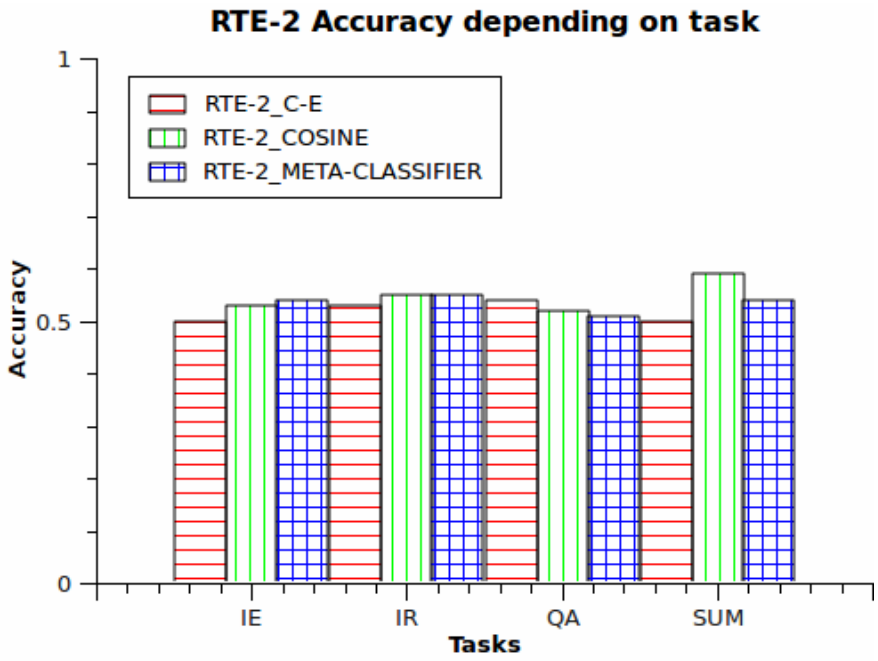

Figure 4.12: RTE-2 comparison with previous results by tasks

Following we present the results over the RTE-3 development data set (coverage) and test data set.

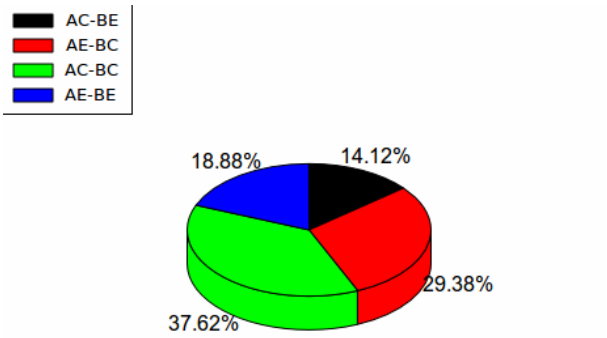

Figure 4.13: RTE-3 meta-classifier coverage

Table 4.27: RTE-3 contingency table

	true	false
true	314	230
false	96	160

Table 4.28: RTE-3 evaluation measures

Accuracy	Precision	Recall	F-measure
0.59	0.57	0.76	0.65

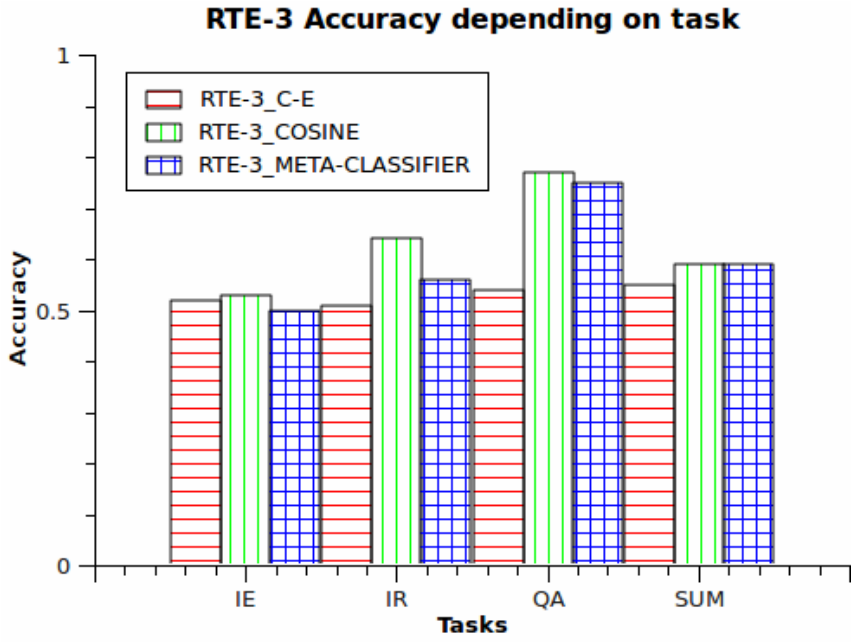

Figure 4.14: RTE-3 comparison with previous results by tasks

4.2.6.2. Experiment 2

In this subsection we present the results of the second meta-classifier over the RTE Challenge. In the RTE-1 and RTE-2the results did not achieve great differences against the Experiment 1. Thus in the RTE-3 the system achieve the best accuracy of all our experiments with 0.61.

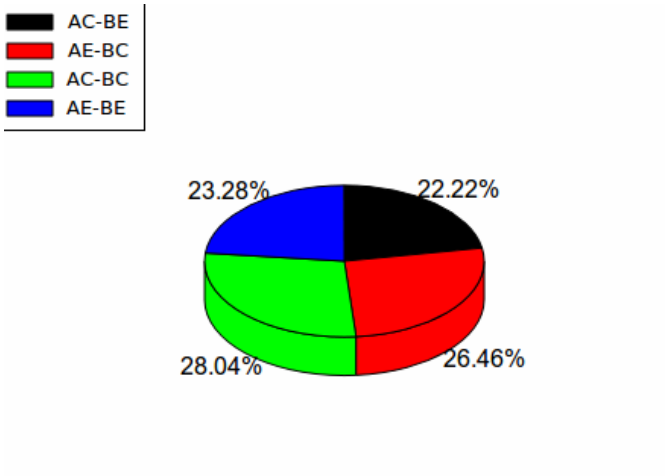

Figure 4.15: RTE-1 meta-classifier coverage

Table 4.29: RTE-1 contingency table

	true	false
true	235	202
false	165	198

Table 4.30: RTE-1 evaluation measures

Accuracy	Precision	Recall	F-measure
0.54	0.53	0.58	0.56

The results showed an improvement using a meta-classifier than only use the non-symmetric measure by itself.

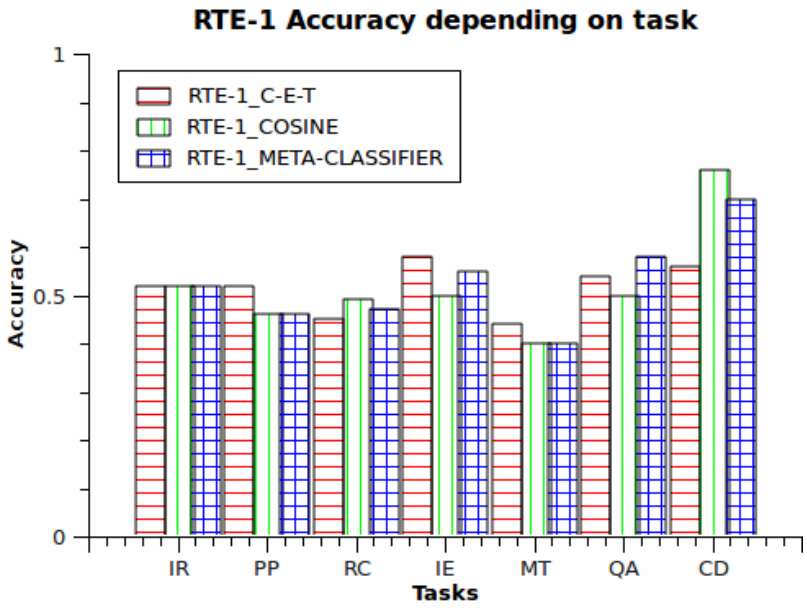

Figure 4.16: RTE-1 comparison with previous results

In Figure 4.16 it is shown the meta-classifier only outperform in the QA task.

The next tables show the results over the RTE-2 Challenge.

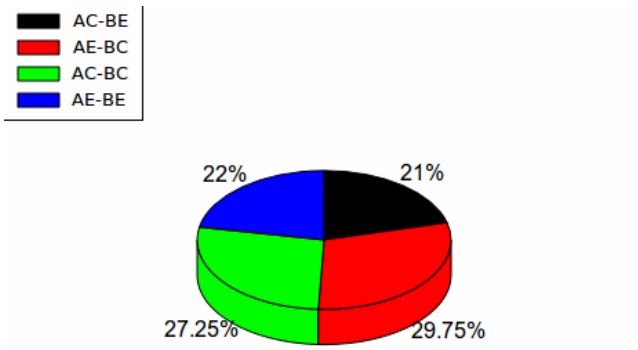

Figure 4.17: RTE-2 meta-classifier coverage

Table 4.31: RTE-2 contingency matrix

	true	false
true	227	208
false	173	192

Table 4.32: RTE-2 evaluation measures

Accuracy	Precision	Recall	F-measure
0.52	0.52	0.56	0.54

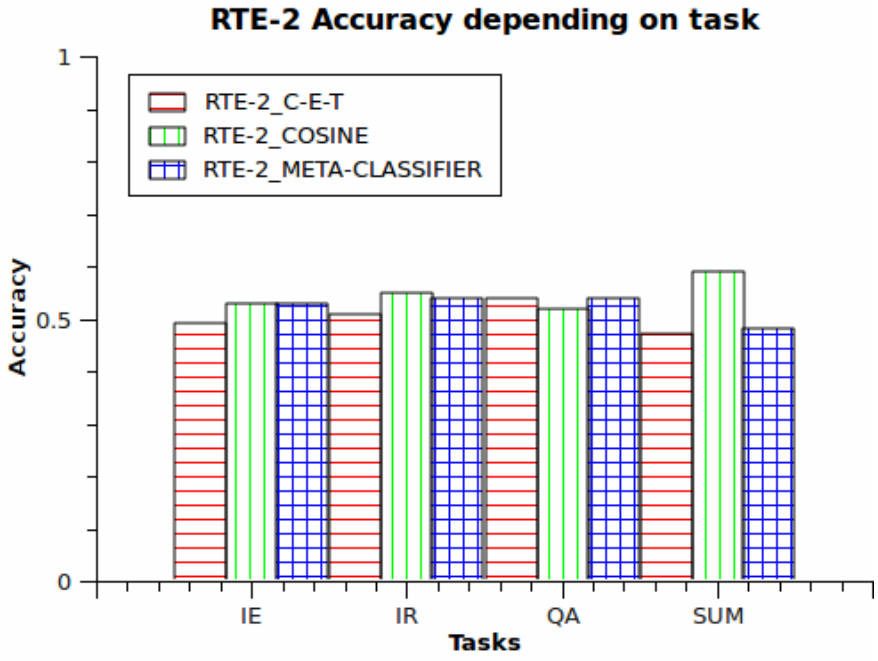

Figure 4.18: RTE-2 comparison with previous results by tasks

In the RTE-3 we achieve the better results for our approach, comparing it to the other results in our research. Thus the results to the RTE-3 were competitive to other participants in the same Challenge.

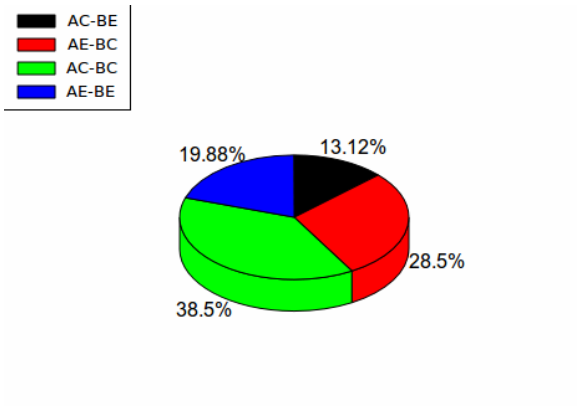

Figure 4.19: RTE-3 meta-classifier coverage

Table 4.33: RTE-3 contingency matrix

	true	false
true	264	163
false	146	227

Table 4.34: RTE-3 evaluation measures

Accuracy	Precision	Recall	F-measure
0.61	0.61	0.64	0.63

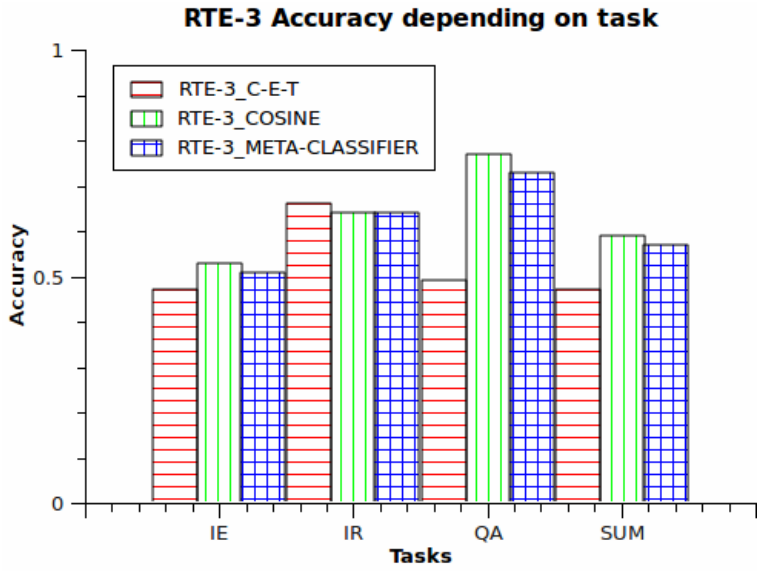

Figure 4.20: RTE-3 comparison with previous results by tasks

4.3. Conclusion

We proposed a non-symmetric similarity measure to the RTE. Therefore our method is unsupervised which is no language dependent.

We have shown that our measure has a lower accuracy than the state of the art methods and outperforms the RTE baseline. These results are significant because they are based on a very simple algorithm that relies on co-occurrences of causal pairs.

We once more confirmed that the web could be used as a lexical resource for RTE.

In our future work we will explore the use of different meta-features for the meta-classifier, as well as linguistically-motivated meta-features (such as a syntactic unit) and evaluate our method against the RTE machine learning approaches.

5. Conclusions

In this thesis we used a variant of the Lesk algorithm for the WSD task based on web counts collected with a search engine. We proposed a new non-symmetric similarity measure for the RTE based on word counts collected from corpus of causal pairs.

We have shown that our variant outperforms some Lesk based methods and outperforms the top unsupervised methods of the Senseval-2 exercise. These results are significant because they are based on a very simple algorithm that relies on co-occurrences scores to the senses of a target word.

We have also shown that our non-symmetric measure has a lower accuracy than the state of the art methods. So our method outperforms the RTE baseline.

We once more confirmed that the web could be used as a lexical resource for WSD and RTE.

In our future work we will explore the use of a WSD method in a RTE approach. Our hypothesis is based on the Yarowsky algorithm (Yarowsky, 1995). The Yarowsky algorithm is an unsupervised learning algorithm for WSD that uses the "one sense per collocation" and the "one sense per discourse" properties of human languages for word sense disambiguation. From observation, words tend to exhibit only one sense in most given discourse and in a given collocation. Therefore if we treat the T-H pair as parts of the same discourse they will exhibit one sense per discourse property. This property will be useful to entailment decision.

5.1. Contributions

The main contributions of this thesis are:

- A variant to the Lesk algorithm for WSD and its software implementation, which outperforms some Lesk based methods and outperforms the top unsupervised methods of the Senseval-2 exercise.
- New symmetric similarity measure for RTE and its software implementation, which outperforms the baseline system.
- A meta-classifier based on a symmetric measure and a non-symmetric measure, which has a competitive accuracy.
- A system for RTE algorithm evaluation.
- A database of the RTE Challenges.
- A database of cause-effect pairs.

5.2. Publications

- Miguel Angel Ríos Gaona, Salvador Godoy Calderón, Alexander Gelbukh. Word Sense Disambiguation with the KORA-Ω Algorithm. Advances in Intelligent and Information Technologies. Special issue of J. Research in Computing Science, ISSN 1870-4069, N 38, 2008, pp. 263–270.
- Miguel Angel Ríos Gaona, Alexander Gelbukh, Sivaji Bandyopadhyay. Web-based Variant of the Lesk Approach to Word Sense Disambiguation. MICAI 2009. Proceedings of 2009 Eighth Mexican International Conference on Artificial Intelligence, ISBN 978-0-7695-3933-1, IEEE CS Press, 2009, pp. 103–107.
- Miguel Angel Ríos Gaona, Alexander Gelbukh, Sivaji Bandyopadhyay. Recognizing Textual Entailment with Statistical Methods. MCPR 2010, 2nd Mexican Conference on Pattern Recognition (to be published).

- Miguel Angel Ríos Gaona, Alexander Gelbukh, Sivaji Bandyopadhyay. Recognizing Textual Entailment Using a Machine Learning Approach. MICAI 2010. Proceedings of 2010 9th Mexican International Conference on Artificial Intelligence (in review).

References

Agirre, E., D. Martinez. (2003). Exploring Automatic Word Sense Disambiguation with Decision Lists and the Web. In: Proc. of the COLING-2000.

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.

Bar-Haim, R., I. Dagan, I. Greental, I. Szpektor, and M. Friedman. (2007). Semantic Inference at the Lexical-Syntactic Level for Textual Entailment Recognition. In Proceedings of the ACLPASCAL Workshop on Textual Entailment and Paraphrasing. Pages 1-9. 28-29 June, Prague,Czech Republic.

Bayer, S., J. Burger, L. Ferro, J. Henderson, A. Yeh. (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.

Banerjee, S. and T. Pedersen. (2002). An adapted Lesk algorithm for word sense disambiguation using WordNet, CICLING 2002.

Bunescu, R.(2003): Associative Anaphora Resolution: A Web-Based Approach. In: Proc. of the EACL-2003 Workshop on the Computational Treatment of Anaphora, Budapest, Hungary.

Brill, E., J. Lin, M. Banko, S. Dumais, A. Ng. (2001): Data-intensive Question Answering. In: Proc. of the Tenth Text Retrieval Conference TREC-2001.

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.

Bos, J. and K. Markert. (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.

Bos, J., S. Clark, M. Steedman, J. Curran, and J. Hockenmaier. (2004). Wide-coverage semantic representations from a ccg parser. In Proc of the 20th International Conference on Computational Linguistics; Geneva, Switzerland.

Clark, P., W.R. Murray, J. Thompson, P. Harrison, J. Hobbs, Fellbaum. (2007). On the Role ofLexical and World Knowledge in RTE3. In Proceedings of the ACL-PASCAL Workshop onTextual Entailment and Paraphrasing. Pages 54-59. 28-29 June, Prague, Czech Republic.

Clark, P. and P. Harrison. (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.

Chambers, N., D. Cer, T. Grenager, D. Hall, C. Kiddon, B. MacCartney, M. C. Marneffe, D. Ramage, E. Yeh, C. Manning. (2007). Learning Alignments and Leveraging Natural Logic. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Pages165-170. 28-29 June, Prague, Czech Republic.

Chierchia, Gennaro and S. McConnell-Ginet. (2000). Meaning and grammar (2nd ed.): an introduction to semantics. MIT Press, Cambridge, MA, USA.

Chklovski, T. and P. Pantel. (2004). Verbocean: Mining the web for fine-grained semantic verbrelations. In Proceedings of EMNLP 2004, Pages 33–40, Barcelona, Spain, July.Association for Computational Linguistics.

Daelemans, W., J. Zavrel, K. Van der Sloot, and A. Van den Bosch. (1998). Timbl: Tilburg memory based learner, version 1.0, reference guide.

Dagan, I. and O. Glickman. (2004). Probabilistic textual entailment: Generic applied modeling of language variability. PASCAL workshop on Text Understanding and Mining.

Dagan, I., O. Glickman, and B. Magnini. (2005). The pascal recognising textual entailment challenge. Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment.

Deerwester, S., S. Dumais, T. Furna, G.W. Landauer, and T. K. Harshman. (1990). Indexing by Latent Semantic Analysis. In Journal of the American Society for Information Science.

Delmonte, R., S. Tonelli, M. Aldo Piccolino Boniforti, A. Bristot, E. Pianta. (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,Southampton, U.K.

Cabrio, E., M. Kouylekov, and B. Magnini. (2008). In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, 2008. Gaithersburg, Maryland, USA.

Fellbaum, C. (1998). WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Mass.

Ferrés, D., and H. Rodríguez. (2007). Machine Learning with Semantic-Based Distances Between Sentences for Textual Entailment. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Pages60-65. 28-29 June, Prague, Czech Republic.

Gonzalo, J., F. Verdejo, I. Chugar. (2003). The Web as a Resource for WSD. In: 1st MEANING Workshop, Spain.

Grefenstette, G. (1999): The World Wide Web as a resource for example-based Machine Translation Tasks. In: Proc. of Aslib Conference on Translating and the Computer. London.

Glickman, O., I. Dagan, and M. Koppel. (2006). A lexical alignment model for probabilistic textual entailment. In Quinonero-Candela, J.; Dagan, I.; Magnini, B.; d'Alché-Buc, F. (Eds.) Machine Learning Challenges. Lecture Notes in Computer Science, Vol. 3944, Pages 287-298, Springer.

Harabagiu, S., M. Pasca, and S. Maiorano. (2000). Experiments with Open-Domain Textual Question Answering. COLING 2000.

Herrera, J., A. Peas, and F. Verdejo. (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. and J. Bensley. (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.

Iftene, A., A. Balahur-Dobrescu. (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.

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

Jijkoun, V. and M. de Rijke. (2005). Recognising Textual Entailment Using Lexical Similarity. Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment.

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

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

Kilgarriff, A. J. et Rosenzweig. (2000). Framework and Results for English SENSEVAL, Computers and the Humanities, 34, (pp. 15-48).

Kouylekov, M. and B. Magnini. (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.

Kozareva, Z. and A. Montoyo. (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, Italy.

Lesk, M. (1986). Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone, Proceedings of SIGDOC.

Li, B., J. Irwin, E.V. Garcia, and A. Ram. (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.

Litkowski, K. C. (2002). Sense Information for Disambiguation: Confluence of Supervised and Unsupervised Methods, Proceedings of the SIGLEX / SENSEVAL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions, Philadelphia.

Lin, C. and E. Hovy. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In Proceedings of the Human Technology Conference 2003 (HLT-NAACL-2003).

Manning, C., and H. Schütze. Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA.

McCarthy, D., J. R. Koeling, and J. Carroll. (2004) Finding predominant senses in untagged text. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics. Barcelona, Spain. pp 280–287.

Mihalcea, R., D.I. Moldovan. (1999): An Automatic Method for Generating Sense Tagged Corpora. In: Proc. of the 16th National Conf. on Artificial Intelligence. AAAI Press.

Mihalcea, R., P. Tarau, E. Figa. (2004). PageRank on Semantic Networks with Application to Word Sense Disambiguation, COLING 2004.

Miller, G. 1991. WordNet: An on-line lexical database. International Journal of Lexicography.

Montalvo-Huhn, O., S. Taylor (2008). Textual Entailment - Fitchburg State College. In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, 2008. Gaithersburg, Maryland, USA.

Pazienza, M.T., M. Pennacchiotti, and F. M. Zanzotto. (2005). Textual Entailment as Syntactic Graph instance: 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.

Resnik, P., and D. Yarowsky. (1997). A perspective on word sense disambiguation. In Proceedings if ACL Siglex Workshop on Tagging With Lexical Semantics, Why, What and How? Washington DC, April.

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

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

Raina, R., A. Haghighi, C. Cox, J. Finkel, J. Michels, K. Toutanova, B. MacCartney, M.C. Marneffe, C. Manning, A. Ng. (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.

Gaizauskas, R., Y. Wilks (1998). Information Extraction: Beyond Document Retrieval. University of Sheffield, Computer Science Dept. Memoranda in Computer and Cognitive Science, CS-97-10.

Pradhan, S., E. Dmitriy and M. Palmer. (2007). SemEval-2007 Task 17: English Lexical Sample, SRL and All Words". Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pp 87–92, Prague, June 2007. Association for Computational Linguistics.

Santamaria, C., J. Gonzalo, and F. Verdejo. (2003): Automatic Association of WWW Directories to Word Senses. Computational Linguistics (2003), Vol. 3, Issue 3 – Special Issue on the Web as Corpus, 485–502.

Siblini, R., and L. Kosseim. (2008). Using Ontology Alignment for the TAC RTE 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.

Tatu, M., and D. Moldovan. (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.

Tatu, M., B. Iles, J. Slavick, A. Novischi, and D. Moldovan. (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, Italy.

Vanderwende, L., D. Coughlin, and B. Dolan. (2005). What Syntax can Contribute in Entailment Task. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages. 13-16, 11–13 April, 2005, Southampton, U.K.

Vasilescu, F., P. Langlais, and G. Lapalme. (2004).Evaluating variants of the Lesk approach for disambiguating words, LREC 2004.

Volk, M. (2002). Using the Web as Corpus for Linguistic Research. In: Catcher of the Meaning. Pajusalu, R., Hennoste, T. (Eds.). Dept. of General Linguistics 3, University of Tartu, Germany.

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

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. ACL 1995.