# DLSITE-1: Lexical Analysis for Solving Textual Entailment Recognition

Óscar Ferrández, Daniel Micol, Rafael Muñoz, and Manuel Palomar

Natural Language Processing and Information Systems Group
Department of Computing Languages and Systems
University of Alicante
San Vicente del Raspeig, Alicante 03690, Spain
{ofe, dmicol, rafael, mpalomar}@dlsi.ua.es

**Abstract.** This paper discusses the recognition of textual entailment in a text-hypothesis pair by applying a wide variety of lexical measures. We consider that the entailment phenomenon can be tackled from three general levels: lexical, syntactic and semantic. The main goals of this research are to deal with this phenomenon from a lexical point of view, and achieve high results considering only such kind of knowledge. To accomplish this, the information provided by the lexical measures is used as a set of features for a Support Vector Machine which will decide if the entailment relation is produced. A study of the most relevant features and a comparison with the best state-of-the-art textual entailment systems is exposed throughout the paper. Finally, the system has been evaluated using the *Second PASCAL Recognising Textual Entailment Challenge* data and evaluation methodology, obtaining an accuracy rate of 61.88%.

## 1 Introduction

Textual Entailment has been proposed recently as a generic framework for modeling semantic variability in many Natural Language Processing (NLP) applications. An entailment relation between two text snippets (text-hypothesis pair) is produced when the hypothesis' meaning can be inferred from the text's.

Some examples of NLP applications that need to detect when the meaning of a text can be inferred from another one could be the followings. In a Question Answering (QA) system, the same answer could be expressed in different syntactic and semantic ways, and a textual entailment module could help such system to identify the forecast answers that entail the expected one. In other applications such as Information Extraction (IE), the textual entailment tool is applied to different variants that express the same relation. In multi-document summarization (SUM), for instance, we could use such tool to extract the most informative sentences, omitting the redundant information. In general, a textual entailment tool would be useful in order to obtain a better performance in a wide range of NLP applications.

Recognising entailment relations is a very complex task that integrates many levels of linguistic knowledge [2] (i.e. lexical, syntactic and semantic levels). Such

complexity has been proven in the two editions of the *PASCAL Recognising Textual Entailment (RTE) Challenge*[1] [6, 3]. These editions of the *PASCAL RTE* have introduced a common task and evaluation framework for textual entailment, covering a broad range of semantic-oriented inferences needed for practical tasks such as the aforementioned applications (concretely QA, IE, Information Retrieval (IR) and SUM). The systems that participated in the challenges used different strategies that combined a wide variety of NLP techniques in order to detect textual entailment. For instance, it is clearly stated that the use of n-grams and subsequence overlap [12, 5], syntactic matching [8], logical inference [4, 13] and Machine Learning (ML) classification [4, 1] is quite appropriate for identifying entailment inferences.

In this paper we propose a system, which we have called *DLSITE-1*, to determine entailment relations based on a wide variety of lexical similarity measures. The aim of using only lexical measures is to achieve a reliable system without need of syntactic and semantic knowledge. Once we have a robust system considering lexical similarities, we will be able to add syntactic and semantic knowledge to it.

The remainder of this paper is structured as follows. The second section details our system and the lexical similarity measures used. The third one illustrates the performed experiments and includes a discussion about the results. Finally, the fourth and last section presents the conclusions of our research and proposes future work.

## 2 System Description

Our system computes the extraction of several lexical measures from the text-hypothesis pairs, which allow us to determine if the entailment relation is produced. Such measures are basically based on word co-occurrences in both the hypothesis and the text, as well as the context where they appear.

Prior to the calculation of the measures, all texts and hypothesis are tokenized and lemmatized. Later on, a morphological analysis is performed as well as a stemmization[2], in order to obtain both the grammatical category and the stem for each word belonging to the two snippets. Once these steps are completed, we are able to create several data structures containing the tokens, stems, lemmas, functional[3] words and the most relevant[4] ones corresponding to the text and the hypothesis. Furthermore, having these structures will allow us to know which of them are more suitable to recognize entailment.

In the following paragraphs we describe in detail the measures applied to the data structures obtained from the previous analysis.

[1] http://www.pascal-network.org/Challenges/RTE/ and http://www.pascal-network.org/Challenges/RTE-2/

[2] We use a Porter stemmer implementation.

[3] As functional words we consider nouns, verbs, adjectives, adverbs and figures (number, dates, etc.).

[4] Considering only nouns and verbs.

- **Simple matching:** word overlap between text and hypothesis is initialized to zero. If a word (token, stem, lemma or functional word) in the hypothesis appears also in the text, an increment of one unit is added to the final weight. Otherwise, no increment is produced. Finally, this weight is normalized dividing it by the length of the hypothesis, calculated as the number of words, as shown in Equation 1.

$$spMatch = \frac{\sum_{i \in H} match(i)}{|H|} \tag{1}$$

where $H$ is the set of tokens, stems, lemmas or functional words of the hypothesis, and $match(i)$ is computed as follows:

$$match(i) = \begin{cases} 1 & \text{if } \exists \text{ j} \in T \text{ i=j,} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

being $T$ the set of tokens, stems, lemmas or functional words of the text.

- **Levenshtein distance:** it is similar to simple matching. However, in this case we calculate the function $match(i)$ for each element of $H$ as:

$$match(i) = \begin{cases} 1 & \text{if } \exists j \in T Lv(i,j) = 0, \\ 0.9 & \text{if } \nexists j \in T Lv(i,j) = 0 \wedge \\ & \exists k \in T Lv(i,k) = 1, \\ \max\left(\frac{1}{Lv(i,j)} \forall j \in T\right) & \text{otherwise.} \end{cases} \tag{3}$$

where $Lv(i,j)$ represents the Levenshtein distance between $i$ and $j$. In our implementation, the cost of an insertion, deletion or substitution is equal to one and the weight assigned to $match(i)$ when $Lv(i,j) = 1$ has been obtained empirically.

- **Consecutive subsequence matching:** this measure assigns the highest relevance to the appearance of consecutive subsequences. In order to perform this, we have generated all possible sets of consecutive subsequences, from length two until the length in words (tokens, stems, lemmas or functional words depending on the data structure used), from the text and the hypothesis. If we proceed as mentioned, the sets of length two extracted from the hypothesis will be compared to the sets of the same length from the text. If the same element is present in both the text and the hypothesis set, then a unit is added to the accumulated weight. This procedure is applied for all sets of different length extracted from the hypothesis. Finally, the sum of the weight obtained from each set of a specific length is normalized by the number of sets corresponding to this length, and the final accumulated weight is also normalized by the length of the hypothesis in words minus one. This measure is defined as follows:

$$LCSmatch = \frac{\sum_{i=2}^{|H|} f(SH_i)}{|H| - 1} \tag{4}$$

where $SH_i$ contains the hypothesis' subsequences of length $i$. Also, $f(SH_i)$ is defined as follows:

$$f(SH_i) = \frac{\sum_{j \in SH_i} match(j)}{|H| - i + 1} \tag{5}$$

being

$$match(j) = \begin{cases} 1 & \text{if } \exists \text{ k} \in ST_i \text{ k=j,} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

where $ST_i$ is the set that contains the text's subsequences of length $i$.
One should note that this measure does not consider non-consecutive subsequences. In addition, it assigns the same relevance to all consecutive subsequences with the same length. Also, the more length the subsequence has, the more relevant it will be considered.

- **Tri-grams:** two sets containing tri-grams of characters belonging to the text and the hypothesis were created. All the occurrences in the hypothesis' tri-grams set that also appear in the text's will increase the accumulated weight in a factor of one unit. Finally, the calculated weight is normalized dividing it by the total number of tri-grams within the hypothesis.
- **ROUGE measures:** ROUGE measures have already been tested for automatic evaluation of summaries and machine translation [10, 9]. For this reason, and considerifng the impact of n-gram overlap metrics in textual entailment, we believe that the idea of integrating these measures in our system is very appeal. We have implemented these measures as defined in [9]. Next, we will proceed to explain them.
  - **ROUGE-N:** determines an n-gram recall between a candidate hypothesis and the reference text. It is computed as follows:

$$ROUGE - N = \frac{\sum_{gram_n \in H} Count_{match}(gram_n)}{\sum_{gram_n \in H} Count(gram_n)} \tag{7}$$

    where $n$ indicates the length of the n-gram ($gram_n$), $Count_{match}(gram_n)$ is the maximum number of n-grams that appear in both the hypothesis and the text, and $Count(gram_n)$ is the number of n-grams within the hypothesis. In our approach, the n-grams are created from the tokens, stems, lemmas and functional words extracted from the text and the

hypothesis, and a set of previous experiments determined that the most suitable values for $n$ are two and three.

- **ROUGE-L:** prior to calculating this measure, we obtained the longest common subsequence (LCS) between the hypothesis and the text, defined as $LCS(T, H)$. The LCS problem consists in finding the longest sequence which is a subsequence of all sequences in a set of sequences[5]. Later on, we applied an LCS-based F-measure to estimate the similarity rate as follows:

$$\begin{aligned} R_{LCS} &= \frac{LCS(T,H)}{|T|} \\ P_{LCS} &= \frac{LCS(T,H)}{|H|} \\ F_{LCS} &= \frac{\left(1+\beta^2\right) \cdot R_{LCS} \cdot P_{LCS}}{R_{LCS} + \beta^2 \cdot P_{LCS}} \end{aligned} \tag{8}$$

  where $\beta = 1$, and $T$ and $H$ are the sets that contain the tokens, stems, lemmas or functional words corresponding to the text and the hypothesis.
- **ROUGE-W:** is quite similar to the ROUGE-L measure. The difference relies on the extension of the basic LCS. ROUGE-W uses a weighted LCS between the text and the hypothesis, $WLCS(T, H)$. This modification of LCS memorizes the length of consecutive matches encountered considering them as a better choice than longer non-consecutive matches. We computed the F-measure based on WLCS as follows:

$$\begin{aligned} R_{LCS} &= f^{-1}\left(\frac{WLCS(T,H)}{f\left(|T|\right)}\right) \\ P_{LCS} &= f^{-1}\left(\frac{WLCS(T,H)}{f\left(|H|\right)}\right) \\ F_{LCS} &= \frac{\left(1+\beta^2\right) \cdot R_{LCS} \cdot P_{LCS}}{R_{LCS} + \beta^2 \cdot P_{LCS}} \end{aligned} \tag{9}$$

  where $f^{-1}$ is the inverse function of $f$. One property that $f$ must have the is that $f(x+y) > f(x) + f(y)$ for all positive integer values[6]. In our experiments we used $f(k) = k^2$, $f^{-1}(k) = k^{1/2}$ and $\beta = 1$.
- **ROUGE-S:** this measure is based on skip-ngrams. A skip-ngram is any combination of $n$ words in their sentence order, allowing arbitrary gaps. ROUGE-S measures the overlap of skip-ngrams between the hypothesis and the text, $SKIP_n(T, H)$. As the aforementioned ROUGE measures, we compute the ROUGE-S-based F-measure as follows:

[5] Definition extracted from http://www.wikipedia.org/

[6] This property ensures that consecutive matches has more scores than non-consecutive matches

$$
\begin{aligned}
R_{LCS} &= \frac{SKIP_n(T,H)}{C\left(|T|,n\right)} \\
P_{LCS} &= \frac{SKIP_n(T,H)}{C\left(|H|,n\right)} \\
F_{LCS} &= \frac{\left(1+\beta^2\right)\cdot R_{LCS}\cdot P_{LCS}}{R_{LCS}+\beta^2\cdot P_{LCS}}
\end{aligned}
\tag{10}
$$

where $\beta = 1$, $C$ is a combinational function and $n$ is the length of the selected skip-gram. For our experiments we developed skip-bigrams and skip-trigram ($n = 2$ and $n = 3$), due to the fact that higher values of $n$ produced meaningless skip-ngrams.

The whole system's architecture is shown in Figure 1. It illustrates how the different modules interact between them as well as the ML algorithm used to decide whether there is entailment or not. Different ML classifiers were considered, being the Support Vector Machine (SVM) the best one for our needs. We have used the SVM implementation of Weka [14], considering each lexical measure as a feature for the training and test stages.

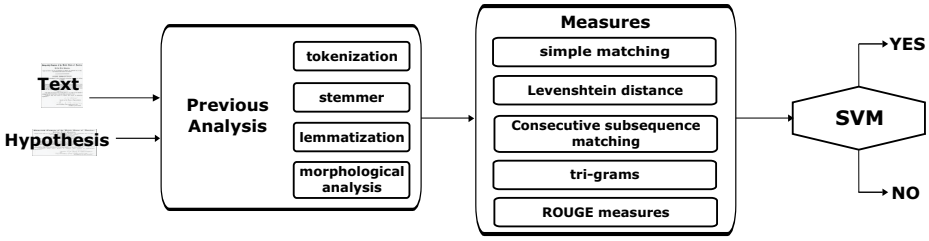


**Fig. 1.** *DLSITE-1* system architecture.

## 3 Experiments and Discussion

The aim of the performed experiments is to check whether our research on lexical measures as a SVM classifier features can achieve satisfactory results considering that only lexical information is used. In this section we present the evaluation environment and the different sets of features obtained applying a selection process. Later on, we show and analyze the results obtained.

### 3.1 Evaluation Environment

To evaluate our system we believe that it is appropriate to use the corpora from the two editions of *PASCAL RTE Challenge*. The organizers of this challenge

provide participants with development and test corpora, both of them containing 800 sentence pairs (text and hypothesis) manually annotated for logical entailment. It is composed of four subsets, each of which corresponds to typical success and failure settings in different tasks, such as Information Extraction (IE), Information Retrieval (IR), Question Answering (QA), and Multi-document Summarization (SUM). For each task, the annotators selected positive entailment examples (annotated YES), as well as negative examples (annotated NO) where entailment is not produced (50%-50% split). The judgments returned by the system will be compared to those manually assigned by the human annotators. The percentage of matching judgments will provide the *accuracy* of the system, i.e. the percentage of correct responses.

Regarding our system's training stage, we used the development corpus from the first and second edition of RTE, namely RTE-1 and RTE-2, respectively. However, the evaluations were performed using only the test corpus provided in RTE-2. The use of the two development corpora increased the number of significant examples in the training data, and, therefore, also increased the final accuracy rate.

### 3.2 Feature Selection

The lexical measures implemented in our system provide a set of 45 features. They have been applied to the text-hypothesis pairs, and, concretely, to their respective words, stems and lemmas. In addition, there are two kinds of lexical measures: those that consider only functional words, and those that only take into account nouns and verbs. The mentioned features were processed as a pool of potentially useful features.

In order to select the best features for our system's purpose, we performed a top-down strategy starting with all available features and iteratively removing one of them in each iteration. The removal criterium was the one that had the lowest information gain. The best feature sets generated using the mentioned strategy were the followings:

- **all_features:** initial set containing all features (*simple matching*, *Levenshtein distance*, *Consecutive subsequence matching*, *Tri-grams* and *ROUGE measures* considering tokens, stems, lemmas and functional words extracted from the text and the hypothesis).
- **R1set:** removing from the all_features set the ones obtained by the $ROUGE-S$ measure (when $S = 2$ and $S = 3$).
- **R2set:** R1set without considering the feature derived from the $ROUGE-L$ and $ROUGE-W$ measures.
- **R3set:** R2set but *the simple matching*, *Levenshtein distance*, *Consecutive subsequence matching*, *Tri-grams* and $ROUGE-N$ measures were only applied to tokens, stems and lemmas extracted from the text and the hypothesis.

### 3.3 Result Analysis

Table 1 summarizes the results obtained with a 10-fold cross validation over the development data and the final system's accuracy using the test corpus provided by RTE-2.

**Table 1.** *Results obtained by the PASCAL RTE-2 evaluation script.*

| | 10-fold Cross Validation | Accuracy (test data) | | | | |
|---|---|---|---|---|---|---|
| | overall | overall | IE | IR | QA | SUM |
| $SVM_{all\_feautures}$ | 0.5941 | 0.6062 | 0.5250 | 0.6050 | 0.5400 | 0.7550 |
| $SVM_{R1set}$ | 0.5897 | 0.6062 | 0.5250 | 0.6000 | 0.5450 | 0.7550 |
| $SVM_{R2set}$ | 0.5919 | 0.6088 | 0.5300 | 0.6150 | 0.5400 | 0.7500 |
| $SVM_{R3set}$ | 0.6013 | **0.6188** | 0.5300 | 0.6300 | 0.5550 | 0.7600 |

As we can observe in the previous table, the differences between feature sets are reduced, being *R3set* the one that achieves better results in both the development and test corpus sets. This fact reveals that the least significant features are produced by the ROUGE measures (except ROUGE-N). In addition, the application of lexical measures to tokens, stems and lemmas obtain better performance than considering functional words or only nouns and verbs.

According to the performed feature analysis and the information gain provided by each one in the training phase, we can deduce that the most significant lexical measures were *Consecutive subsequence matching* and *tri-grams* applied to the tokens and lemmas extracted from the text-hypothesis pair. One should note that these statements depend on the idiosyncrasies of the RTE corpora. However, these corpora are, nowadays, the most reliable for evaluating textual entailment systems.

On the other hand, the fact that the proposed system only uses lexical information reduces its capability to recognise entailment relation. One example could be the pair number 38 from the RTE-2 test corpus, which is shown as follows:

**Text:** Considering the amount of rain that soaked Riviera, Campbell didn't expect to complete his second round Friday in the Nissan Open.

**Hypothesis:** Campbell finished his second round Friday.

In this case, the hypothesis' subsequences "*Campbell*" and "*his second round Friday*" match exactly with the text, producing a high lexical similarity value. The lack of semantic knowledge causes that the system suggests true entailment even although the entailment relation does not exist. This deficiency could be solved adding modules that deal with synonyms and negations contributing to establish different meaning to the text and the hypothesis.

Finally, a comparison between the RTE-2 participating systems is exposed. Table 2 shows the results that *DLSITE-1* such systems obtained in the RTE-2 Challenge.

**Table 2.** *Comparative evaluation within the RTE-2 environment.*

| System | Accuracy (test data) | | | | |
|---|---|---|---|---|---|
| | overall | IE | IR | QA | SUM |
| (Hickl et. al, 2006) [7] | 0.7538 | 0.7300 | 0.7450 | 0.6950 | 0.8450 |
| (Tatu et. al, 2006) [13] | 0.7375 | 0.7150 | 0.7400 | 0.7050 | 0.7900 |
| (Zanzotto et. al, 2006) [15] | 0.6388 | – | – | – | – |
| (Adams, 2006) [1] | 0.6262 | 0.505 | 0.595 | 0.685 | 0.720 |
| **DLSITE-1** | 0.6188 | 0.5300 | 0.6300 | 0.5550 | 0.7600 |
| (Bos et. al, 2006) [4] | 0.6162 | 0.505 | 0.660 | 0.565 | 0.735 |
| ... | | | | | |
| (Ferrández et. al, 2006) [11] | 0.5563 | 0.4950 | 0.5800 | 0.6100 | 0.5400 |

As we can see in Table 2, *DLSITE-1* would have reached the fifth place in the RTE-2 ranking, out of twenty four participants.

The baseline we set for our system was to achieve better results than the ones we obtained with our last participation in RTE-2 (see [11], last row in Table 2). As stated in [11], our previous system obtains a semantic similarity score by means of logic forms derived to the dependency trees from the pair text-hypothesis and WordNet. However, although its results were promising we desired to improve them tackling the recognision of textual entailment from a concrete setting (in this case a lexical setting). This approach allows us to achieve good result considering only lexical information and, subsequently add others kinds of information (e.g. syntactic and semantic) in order to improve the system.

In addition, we would like to emphasize the fact that all systems shown in Table 2 used more knowledge than the information which could be provided by lexical measures. For example, in [1] the author uses WordNet in order to obtain the lexical relation between two tokens as well as a negation detector. The approach in [7] combines lexico-semantic information obtained by a large collection of paraphrases and NLP applications (e.g. named entity recognition, temporar/spatial normalization, semantic role labeling, coreference, etc.). Finally, the system exposed in [13] contains a knowledge representation based on a logic proving setting with NLP axioms.

## 4 Conclusions

The main contribution of this research is the development of a system for solving textual entailment relations considering only lexical information. To achieve this, we implemented and applied a wide variety of lexical measures. The reason why we make use of this amount of measures and information is motivated by the fact that the integration of more complex semantic knowledge is a delicate task as it is demonstrated by the amount of work developed in the last years. Therefore, our goal is to develop a robust system without complex syntactic-

semantic knowledge. Such expertise may be added to our approach in a near future.

In a nutshell, *DLSITE-1* is a textual entailment system that deals with the entailment phenomenon from a lexical point of view, applying relevant lexical measures to deduce entailment relations. It successfully overcomes the RTE task achieving overall accuracy rates higher than 61%. Based on this, the authors of this paper believe that it is easier to perform the recognition task in three separate levels (lexical, syntactic and semantic) and, afterwards, combine them into a complete system.

As of future work, we are interested in improving our system investigating the addition of syntactic and semantic knowledge. Due to the fact that the system achieves high accuracy rates considering only lexical similarities, the next step would be to integrate different tools and strategies to add other kinds of knowledge, such as syntactic and semantic. For instance, we could use resources to generate syntactic dependency trees and obtain similarities between them, including modules that process synonyms and other semantic relations. In addition, extraction of speech knowledge representations by means of techniques based on named entity recognition, co-references and role labeling could be an important improvement.

Moreover, we would like to emphasize that, although the proposed lexical similarity measures need some language dependent tools (e.g. lemmatizer, stemmer and morphological analyzer), the system could be easily ported to other languages. This research line would represent possible future work as well.

## Acknowledgments

This research has been partially funded by the Spanish Government under project TIN2006-15265-C06-01 and by the QALL-ME consortium, which is a 6th Framenwork Research Programme of the European Union (EU), contract number: FP6-IST-033860. The authors would like to thank the EU for the financial support and the partners within the consortium for a fruitful collaboration. For more information about the QALL-ME consortium, please visit the consortium home page, http://qallme.itc.it/.

## References

1. Rod Adams. Textual Entailment Through Extended Lexical Overlap. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 128–133, Venice, Italy, April 2006.
2. R. Bar-Haim, I. Szpecktor, and O. Glickman. Definition and analysis of intermediate entailment levels. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 55–60, Ann Arbor, Michigan, June 2005.
3. Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The Second PASCAL Recognising Textual Entailment Challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 1–9, April 2006.

4. Johan Bos and Katja Markert. When logical inference helps determining textual entailment (and when it doesn't). In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 98–103, Venice, Italy, April 2006.
5. Daoud Clarke. Meaning as Context and Subsequence Analysis for Entailment. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 134–139, April 2006.
6. Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 1–8, Southampton, UK, April 2005.
7. Andrew Hickl, Jeremy Bensley, John Williams, Kirk Roberts, Bryan Rink, and Ying Shi. Recognizing Textual Entailment with LCC's GROUNDHOG System. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 80–85, Venice, Italy, April 2006.
8. Milen Kouylekov and Bernardo Magnini. Tree Edit Distance for Recognizing Textual Entailment: Estimating the Cost of Insertion. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 68–73, Venice, Italy, April 2006.
9. Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.
10. Chin-Yew Lin and Franz Josef Och. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. In *ACL*, pages 605–612, 2004.
11. Óscar Ferrández, Rafael M. Terol, Rafael Muñoz, Patricio Martínez-Barco, and Manuel Palomar. An approach based on Logic forms and wordnet relationships to textual entailment performance. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 22–26, Venice, Italy, April 2006.
12. Diana Pérez and Enrique Alfonseca. Application of the Bleu algorithm for recognising textual entailments. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 9–12, Southampton, UK, April 2005.
13. Marta Tatu, Brandon Iles, John Slavick, Adrian Novischi, and Dan Moldovan. COGEX at the Second Recognizing Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 104–109, Venice, Italy, April 2006.
14. Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
15. F.M. Zanzotto, A. Moschitti, M. Pennacchiotti, and M.T. Pazienza. Learning textual entailment from examples. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 50–55, Venice, Italy, April 2006.