A Semantic-less Approach for the Textual Entailment Recognition Task

Daniel Micol, Óscar Ferrández, 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
{*dmicol, ofe, rafael, mpalomar*}@*dlsi.ua.es*

Abstract

In this paper we describe the system we have developed to overcome the textual entailment recognition task without any kind of semantic knowledge. For this purpose we have designed and implemented two modules. The first one analyzes the lexical information extracted from the phrases, while the second studies them from a syntactic perspective. The main goal of this research is to allow us to acknowledge the maximum accuracy that we can achieve without a semantic analysis. To evaluate our system we used the test corpus sets from *Second and Third PASCAL Recognising Textual Entailment Challenges*, obtaining accuracy rates of 62.12% and 65.63%, respectively.

Keywords

Textual Entailment, Lexical information, Syntactic information.

1 Introduction

The field of Natural Language Processing is an essential part of Artificial Intelligence that studies the communication and interaction between human beings and computers. Within this area, Textual Entailment has been defined as a generic framework for modeling semantic variability that appears when a concrete meaning is described in different manners. Throughout this paper we will follow the guidelines proposed in *PASCAL Recognising Textual Entailment*[1] (RTE-1, RTE-2 and RTE-3) [1, 2], which establishes that the meaning of a text snippet (termed hypothesis) should be inferred from the meaning of another one (namely text).

This paper discusses an approach that attempts to detect when the entailment is produced, and focuses on determining if such relation appears due to lexical or syntactic implications between the texts. We propose several methods that mainly rely on lexical and syntactic inferences in order to address the entailment recognition task. The reason why we have decided not to use semantic knowledge is because we would like to acknowledge the maximum amount of information that the mentioned two perspectives can provide, so that we will be able to combine them later on with a semantic module in an optimal way.

The remainder of this paper is structured as follows. The second section details the aforementioned methods, and the third one illustrates the performed experiments and includes a discussion about the results. Finally, the last section presents the conclusions of our research and proposes possible future work.

2 Methods

As we previously mentioned, our system is composed of two modules, which we will now explain.

2.1 Lexical approach

This method relies on the computation of a wide variety of lexical measures that basically consist of overlap metrics. Some researchers have already used this kind of metrics [9]. However, the main novelty of our approach is that it does not use semantic knowledge.

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 stemming. Once these steps are completed, we create several data structures that contain the corresponding tokens, stems, lemmas, functional[2] words and the most relevant[3] ones corresponding to the text and the hypothesis. The lexical measures will be applied to these structures and will allow us to determine which of them are more suitable for recognizing entailment situations, depending on the similarity rates that they provide.

We will now describe the lexical measures included in our system. Each of them calculates a similarity value between text and hypothesis that will allow us to determine if there is entailment between both of them or not.

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

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

[3] Considering only nouns and verbs.

2.1.1 Simple matching

Word overlapping between text and hypothesis is initialized to zero. If a word of the hypothesis appears also in the text, an increment of one unit is added to the similarity value. Finally, the weight is normalized dividing it by the length of the hypothesis measured as the number of words.

2.1.2 Levenshtein distance

This distance is similar to simple matching. However, in this case we calculate the value of the function that represents the occurrences in the text of each element that belongs to the hypothesis, denoted by $m(i)$, as defined in Equation 1.

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

where $Lv(i,j)$ represents the Levenshtein distance [4] between i and j. In our implementation, the cost of an insertion, deletion or substitution is equal to one and the weight assigned to $m(i)$ when $Lv(i,j) = 1$ has been obtained empirically.

2.1.3 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, 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 ones 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 to 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 such length, and the final accumulated weight is also normalized dividing it by the length of the hypothesis in words minus one. This measure is defined as shown in Equation 2.

$$CSmatch = \frac{\sum_{i=2}^{|H|} f(SH_i)}{|H| - 1} \quad (2)$$

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

$$f(SH_i) = \frac{\sum_{j \in SH_i} m(j)}{|H| - i + 1} \quad (3)$$

being $m(j)$ equal to one if there exists an element k that belongs to the set that contains the text's subsequences of length i, such that $k = j$.

We would like to point out that this measure does not consider non-consecutive subsequences. In addition, it assigns the same relevance to all consecutive subsequences with the same length. Furthermore, the longer the subsequence is, the more relevant it will be considered in our system.

2.1.4 Tri-grams

Two sets containing tri-grams of letters that belong to the text and the hypothesis were created. All the occurrences of the hypothesis' tri-grams set that also appear in the text's will increase the accumulated weight by a factor of one unit. The calculated weight is then normalized dividing it by the total number of tri-grams within the hypothesis.

2.1.5 ROUGE measures

ROUGE measures have already been tested for automatic evaluation of summaries and machine translation [5, 6]. For this reason, and considering the impact of n-gram overlap metrics in textual entailment, we believe that the idea of integrating these measures[4] in our system is very appealing. We have implemented them as defined in [5].

Within the entire set of measures, each one of them is considered as a feature for the training and test stages of a machine learning algorithm. The selected one was a Support Vector Machine [11] due to the fact that its behavior is suitable for recognizing entailment relations.

Next, we present a true entailment text-hypothesis pair example, and show how the lexical approach calculates the corresponding similarity rate.

> **Text:** *The destruction of the ozone layer was first noticed in the late 1980s as a hole over Antarctica.*
> **Hypothesis:** *The ozone hole was first noticed in the late 1980s.*

The average values obtained for each measure considering tokens, lemmas and content words are the followings:

- Simple matching = 1
- Levenshtein distance = 1
- Consecutive subsequence matching = 0.34
- Tri-grams = 1
- ROUGE measures = from 0.4 using ROUGE-S to 0.66 using ROUGE-L

As it can be observed in the previous example, simple matching, Levenshtein distance and tri-grams achieve the highest possible score, due to the fact that all word occurrences in the hypothesis also appear in the text. However, regarding the consecutive subsequence matching measure, there are some hypothesis' consecutive subsequences that do not appear in the text, but the appearance of the subsequence *"was*

[4] The considered measures were ROUGE-N with n=2 and n=3, ROUGE-L, ROUGE-W and ROUGE-S with s=2 and s=3.

first noticed in the late 1980s" produces that this measure achieves a relatively high score. Finally, ROUGE measures have a similar behavior to the previous one, achieving different scores depending on the type of measures used.

2.2 Syntactic approach

This approach aims to provide a good accuracy rate by using few modules that are based on syntactic knowledge. These include tree construction, filtering, tree embedding detection and tree node matching.

2.2.1 Tree generation

The first module constructs the corresponding syntactic dependency trees. For this purpose, *MINIPAR* [7] output is generated and afterwards parsed for each text and hypothesis of our corpus. Phrase tokens, along with their grammatical information, are stored in an on-memory data structure that represents a tree.

2.2.2 Tree filtering

Once the tree has been constructed, we may want to discard irrelevant data in order to reduce our system's response time and noise. For this purpose we have generated a list of relevant grammatical categories (shown in Table 1) that will allow us to remove from the tree all those tokens whose category does not belong to such list. The resulting tree will have the same structure as the original, but will not contain any stop words nor tokens with minor relevance, such as determinants or auxiliary verbs.

2.2.3 Tree embedding detection

The next step of the syntactic approach consists in determining whether the hypothesis' syntactic dependency tree is embedded into the text's. A tree, T_1, is embedded into another one, T_2, if all nodes and branches of T_1 appear in T_2 as well [3]. Therefore, in this module we attempt to find a match of the hypothesis' syntactic structure within the text's. Since this is a very strict matching process, we will believe that there is entailment if we are able to find a coincidence. Otherwise we will not be able to assure this and will execute the next module of our system, which is described in the following subsection.

2.2.4 Tree node matching

In this stage we proceed to perform a tree node matching process, termed alignment, between both the text and the hypothesis. This operation consists in finding pairs of tokens in both trees whose lemmas are identical, no matter whether they are in the same position within the tree. Some authors have already designed similar matching techniques, such as the ones described in [8, 10]. However, these include semantic constraints that we have decided not to consider. The reason of this decision is that we desired to overcome the textual entailment recognition task from an exclusively syntactic perspective.

Let τ and λ represent the text's and hypothesis' syntactic dependency trees, respectively. We assume we have found a word, namely β, present in both τ and λ. Now let γ be the weight assigned to β's grammatical category (Table 1), σ the weight of β's grammatical relationship (Table 2), μ an empirically calculated value that represents the weight difference between tree levels, and δ_β the depth of the node that contains the word β in λ. We define the function that provides the relevance of a word as follows:

$$\phi(\beta) = \gamma \cdot \sigma \cdot \mu^{-\delta_\beta} \tag{4}$$

The value obtained by calculating this expression would represent the relevance of a word in our system. The experiments performed reveal that the optimal value for μ is 1.1.

Grammatical category	**Weight**
Verbs, verbs with one argument, verbs with two arguments, verbs taking clause as complement	1.0
Nouns, numbers	0.75
Be used as a linking verb	0.7
Adjectives, adverbs, noun-noun modifiers	0.5
Verbs *Have* and *Be*	0.3

Table 1: *Weights assigned to the relevant grammatical categories (empirically calculated).*

Grammatical relationship	**Weight**
Subject of verbs, surface subject, object of verbs, second object of ditransitive verbs	1.0
The rest	0.5

Table 2: *Weights assigned to the grammatical relationships (empirically calculated).*

For a given pair (τ, λ), we define the set ξ as the one that contains all words present in both trees, being $\xi = \tau \cap \lambda \ \ \forall \alpha \in \tau, \beta \in \lambda$. Therefore, the similarity rate between τ and λ, denoted by the symbol ψ, would be as defined in Equation 5.

$$\psi(\tau, \lambda) = \sum_{\nu \in \xi} \phi(\nu) \tag{5}$$

One should note that a requirement of our system's similarity measure would be to be independent of the hypothesis length. Thus, we must define the normalized similarity rate, as shown in Equation 6.

$$\overline{\psi(\tau, \lambda)} = \frac{\sum_{\nu \in \xi} \phi(\nu)}{\sum_{\beta \in \lambda} \phi(\beta)} \tag{6}$$

Once the similarity value has been calculated, it will be provided to the user together with the corresponding text-hypothesis pair identifier. It will be his responsibility to choose an appropriate threshold that will represent the minimum similarity rate to be considered as entailment between text and hypothesis. All

values that are under such a threshold will be marked as not entailed.

We will now show the behavior of the syntactic module for the text-hypothesis pair example shown at the end of section 2.1. For this purpose, we will first generate the corresponding syntactic dependency trees that are shown in Figures 1 and 2.

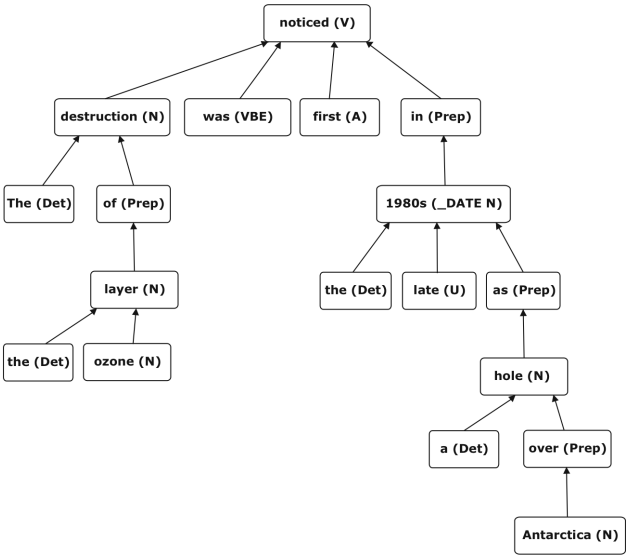

Fig. 1: *The destruction of the ozone layer was first noticed in the late 1980s as a hole over Antarctica.*

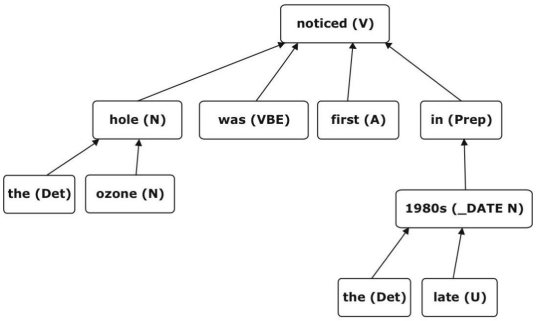

Fig. 2: *The ozone hole was first noticed in the late 1980s.*

The next step would be to perform a tree filtering over the text's and the hypothesis' trees. After this process has been completed, we will try to determine whether there is an entailment relation by calculating the value of function ϕ for each remaining word, based on the values shown in Tables 1 and 2.

$$\phi(noticed) = 1.0 \cdot 1.0 \cdot 1.1^{-1} = 0.91$$
$$\phi(hole) = 0.75 \cdot 1.0 \cdot 1.1^{-2} = 0.62$$
$$\phi(ozone) = 0.75 \cdot 0.5 \cdot 1.1^{-3} = 0.28$$
$$\phi(was) = 0.7 \cdot 0.5 \cdot 1.1^{-2} = 0.29$$
$$\phi(first) = 0.5 \cdot 0.5 \cdot 1.1^{-2} = 0.21$$
$$\phi(1980) = 0.75 \cdot 0.5 \cdot 1.1^{-2} = 0.31$$

Combining all these values we will be able to obtain the similarity rate of the exposed text-hypothesis pair, as $\psi(\tau, \lambda) = \phi(noticed)+\phi(hole)+\phi(ozone)+\phi(was)+\phi(first)+\phi(1980) = 2.62$.

The final step is to calculate the normalized similarity value as defined in Equation 6. However, we would like to point out that in the proposed example all words within the hypothesis also appear in the text. Therefore, the value of the denominator of the fraction from Equation 6 will be the same as the numerator, so the normalized similarity value will be the maximum possible. Since the obtained rate has a high value, we will consider the input pair as entailed.

3 Experimental results

The experimental results shown in this paper were obtained processing a set of text-hypothesis pairs from RTE-2 [1] and RTE-3[5]. The organizers of this challenge provide participants with development and test corpora, both of them with 800 sentence pairs (text and hypothesis) manually annotated for logical entailment. 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.

Table 3 shows the results obtained by both approaches individually (lexical and syntactic) and by combining them. This last approach consists in obtaining the entailment value that achieves the best performance. If both methods, lexical and syntactical, agree, then the judgement is straightforward, but if they disagree we then set the value depending on the performance of each one for true and false entailment situations. In our case, the lexical method performs better while dealing with negative examples, i.e., when there is no entailment relation, so this decision will prevail over the rest. Otherwise, the syntactic one shall decide the judgement.

As we can see in Table 3, the collaborative approach obtains the best results for the RTE-2[6] corpus, but using the RTE-3 corpus the approach that obtained the best performance was the lexical one. This makes us believe that an appropriate combination of these two kinds of knowledge (lexical and syntactic) would improve the entailment recognition. In addition, depending on the target task where the entailment is produced, the lexical approach performs better than the syntactic, and vice versa. For instance, the lexical method performs better when the pair belongs to the IR task. We would like to point out that, at the moment, these statements depend on the idiosyncrasies of the RTE corpora. However, these corpora are, nowadays, the most reliable source for evaluating textual entailment systems.

[5] The *Third PASCAL Recognising Textual Entailment Challenge* has not finished yet. Therefore, we know our individual results although we cannot compare them with the rest of the participating groups.

[6] If our system had participated in the *Second PASCAL Recognising Textual Entailment Challenge*, we would have obtained the fifth position out of twenty-four participating groups.

RTE-2	Development corpus	Test corpus				
	Overall	Overall	IE	IR	QA	SUM
Lexical	0.6013	0.6188	0.5300	0.6300	0.5550	0.7600
Syntactic	0.5750	0.6075	0.5050	0.6450	0.5950	0.6850
Both	**0.6087**	**0.6212**	**0.5100**	**0.6550**	**0.6250**	**0.6950**
RTE-3	Development corpus	Test corpus				
	Overall	Overall	IE	IR	QA	SUM
Lexical	**0.7012**	**0.6563**	**0.5150**	**0.7350**	**0.7950**	**0.5800**
Syntactic	0.6450	0.5925	0.5050	0.6350	0.6300	0.6000
Both	0.6900	0.6375	0.5150	0.7150	0.7400	0.5800

Table 3: *Accuracy rates obtained using the RTE-2 and RTE-3 development and test corpora.*

4 Conclusions and future work

In this paper we have presented a system for detecting textual entailment relations considering mainly lexical and syntactical information. A wide variety of lexical measures as well as syntactic structure comparisons were performed for this purpose. Decomposing the textual entailment task into subtasks allows finer analysis and high accuracy rates as well. As said before, we have been able to build a precise system without need of semantic knowledge, as a difference with most of the current state of the art approaches [1].

The separate analysis of the lexical and syntactic approaches has allowed us to study the maximum amount of knowledge that these perspectives can provide. In addition, we have been able to investigate our system's behavior when both approaches were combined. This is very useful for determining the optimal combination procedure, and will help us to couple a semantic module that does not produce conflicts with the ones described in this paper. We believe that this research line that analyzes the different kinds of knowledge separately allows a more accurate analysis of the successes and failures and the construction of a cleanly designed system.

Regarding future work, we are highly motivated in adding a semantic knowledge module to our system. Huge amounts of work in this line have been carried out by the research community within the last years. To add this kind of knowledge, we propose to extract a high amount of semantic information that would allow us to construct a characterized representation based on the input text, so that we can deduce entailment even if there is no apparent lexical nor syntactic structure similarity between text and hypothesis. This would mean to create an abstract conceptualization of the information contained in the analyzed phrases, allowing us to deduce ideas that are not explicitly mentioned in the parsed text-hypothesis pairs.

In addition, we have observed from the results shown in Table 3 that the accuracy of our system differs between tasks. Thus, we would like to apply different entailment recognition techniques based on the task of the text-hypothesis pair that is being analyzed.

Finally, due to the fact that recognizing textual entailment is a very complex task, we would like to tune the recognition by creating uncertainty thresholds. Such levels would include the situations where the system does not have enough information to determine if there is an entailment relation.

Acknowledgments

This research has been partially funded by the QALL-ME consortium, contract number FP6-IST-033860, and by the Spanish Government under the project CICyT number TIN2006-1526-C06-01. It has also been supported by the undergraduate research fellowships financed by the Spanish Ministry of Education and Science, and the project ACOM06/90 supported by the Spanish Generalitat Valenciana.

References

[1] R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. The Second PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 1–9, Venice, Italy, April 2006.

[2] I. Dagan, O. Glickman, and B. 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.

[3] S. Katrenko and P. Adriaans. Using Maximal Embedded Syntactic Subtrees for Textual Entailment Recognition. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 33–37, Venice, Italy, April 2006.

[4] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, Feburary 1966.

[5] C.-Y. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In S. S. Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the Association for Computational Linguistics Workshop*, pages 74–81, Barcelona, Spain, July 2004.

[6] C.-Y. Lin and F. J. Och. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. In *Association for Computational Linguistics*, pages 605–612, July 2004.

[7] D. Lin. Dependency-based Evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain, 1998.

[8] B. MacCartney, T. Grenager, M.-C. de Marneffe, D. Cer, and C. D. Manning. Learning to recognize features of valid textual entailments. In *Proceedings of the North American Association of Computational Linguistics*, pages 41–48, New York City, New York, United States of America, June 2006.

[9] J. Nicholson, N. Stokes, and T. Baldwin. Detecting Entailment Using an Extended Implementation of the Basic Elements Overlap Metrics. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 122–127, Venice, Italy, April 2006.

[10] R. Snow, L. Vanderwende, and A. Menezes. Effectively using syntax for recognizing false entailment. In *Proceedings of the North American Association of Computational Linguistics*, pages 33–40, New York City, New York, United States of America, June 2006.

[11] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques.* 2nd Edition, Morgan Kaufmann, San Francisco, 2005.