Textual Entailment Recognition Based on Dependency Analysis and *WordNet*

Jesús Herrera, Anselmo Peñas, Felisa Verdejo

Departamento de Lenguajes y Sistemas Informáticos
Universidad Nacional de Educación a Distancia
Madrid, Spain
{jesus.herrera, anselmo, felisa}@lsi.uned.es

Abstract. The Recognizing Textual Entailment System shown here is based on the use of a broad-coverage parser to extract dependency relationships; in addition, *WordNet* relations are used to recognize entailment at the lexical level. The work investigates whether the mapping of dependency trees from text and hypothesis give better evidence of entailment than the matching of plain text alone. While the use of *WordNet* seems to improve system's performance, the notion of mapping between trees here explored (inclusion) shows no improvement, suggesting that other notions of tree mappings should be explored such as tree edit distances or tree alignment distances.

1 Introduction

Textual Entailment Recognition (RTE) aims at deciding whether the truth of a text entails the truth of another text called hypothesis. This concept has been the basis for the PASCAL[1] RTE Challenge [3].

The system presented here is aimed at validating the hypothesis that (i) a certain amount of semantic information could be extracted from texts by means of the syntactic structure given by a dependency analysis, and that (ii) lexico-semantic information such as *WordNet* relations can improve RTE.

In short, the techniques involved in this system are the following:

– Dependency analysis of texts and hypothesises.
– Lexical entailment between dependency tree nodes using *WordNet*.
– Mapping between dependency trees based on the notion of *inclusion*.

For the experiments, the PASCAL RTE Challenge 2005 corpora have been used. Two corpora are available, one for training and a second used to test systems' performance after training. Each corpus is compound by a set of hypothesis and text pairs where the objective is to determine whether the text entails the hypothesis or not for each pair.

In section 2 the architecture of the proposed system is described. Section 3 shows how lexical entailment is accomplished. Section 4 presents the methodology followed

---

[1] Pattern Analysis, Statistical Modeling and Computational Learning Network of Excellence. http://www.pascal-network.org/

to evaluate matching between dependency trees. Section 5 describes the experiments accomplished with the system. In section 6 the results obtained are shown. Finally, some conclusions are given.

2   System's Architecture

The proposed system is based on surface techniques of lexical and syntactic analysis. It works in a non-specific way, not giving any kind of special treatment for the different tasks considered in the Challenge (Comparable Documents, Question Answering, etcetera) [3].

System's components, whose graphic representation is shown in figure 1, are the following:

1. A dependency parser, based on Lin's *Minipar* [9], which normalizes data from the corpus of text and hypothesis pairs and accomplishes the dependency analysis, generating a dependency tree for every text and hypothesis.
2. A lexical entailment module, which takes the information given by the parser and returns the hypothesis' nodes that are entailed by the text. A node is a vertex of the dependency tree, associated with a lexical unit and containing all the information computed by the dependency parser (lexical unit, lemma, part-of-speech, etcetera). This module uses *WordNet* in order to find multiwords and synonymy, similarity, hyponymy, *WordNet*'s entailment and negation relations between pairs of lexical units, as shown in section 3.
3. A matching evaluation module, which searches for paths into hypothesis' dependency tree, conformed by lexically entailed nodes. It works as described in section 4.
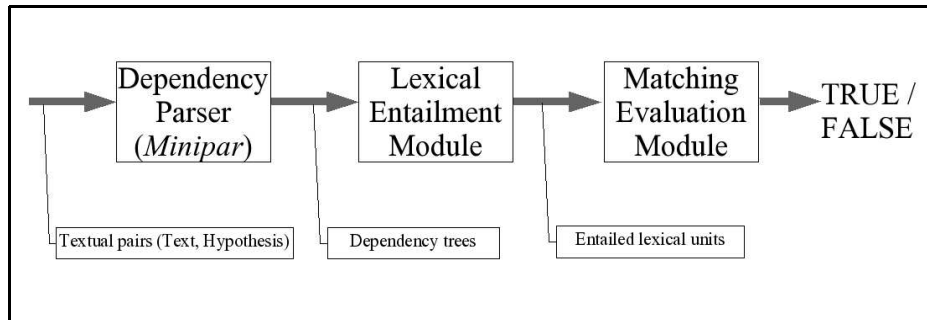
Fig. 1. System's architecture.

The system accepts pairs of text snippets (text and hypothesis) at the input and gives a boolean value at the output: *TRUE* if the text entails the hypothesis and *FALSE* otherwise.

3 Lexical Entailment

A module of lexical entailment is applied over the nodes of both text and hypothesis, as shown in figure 1. This module gets its input from the output of the dependency parser (see figure 1); as described in section 2, the dependency parser provides a dependency tree for every text and hypothesis. The output of the module of lexical entailment is a list of pairs *(T,H)* where *T* is a node in the text tree whose lexical unit entails the lexical unit of the node *H* in the hypothesis tree. This entailment at the word level considers *WordNet* relations, detection of *WordNet* multiwords and negation, as follows:

3.1 Synonymy and Similarity

The lexical unit *T* entails the lexical unit *H* if they can be synonyms according to *WordNet* or if there is a relation of similarity between them. Some examples were found in the PASCAL Challenge training corpus such as, for example: *discover* and *reveal*, *obtain* and *receive*, *lift* and *rise*, *allow* and *grant*, etcetera.

The rule implemented in the lexical entailment module was the following:

– *entails(T, H)* IF *synonymy(T, H)* OR *WN_similarity(T, H)*

As an example, for the lexical units *allow* and *grant*, since *synonymy(allow, grant) is TRUE* then the module determines that *entails(allow, grant)*, i.e., *allow* and *grant* are lexically entailed by a synonymy relation. Another example is given for the lexical units *discover* and *reveal*: since *WN_similarity(discover, reveal) is TRUE*, then the module determines that *entails(discover, reveal) is TRUE*.

3.2 Hyponymy and *WordNet* Entailment

Hyponymy and entailment are relations between *WordNet* synsets having a transitive property. Some examples after processing the training corpus of PASCAL Challenge are: *glucose* entails *sugar*, *crude* entails *oil*, *kill* entails *death*.

The rules implemented were:

– *entails(T, H)* IF exists a synset S_T including *T* and a synset S_H including *H* such as *hyponymy(S_T,S_H)*
– *entails(T, H)* IF exists a synset S_T including *T* and a synset S_H including *H* such as *WN_en-tailment(S_T,S_H)*
– *entails(T, H)* IF exists a path from a synset S_T including *T* to a synset S_H including *H* conformed by hyponymy and/or *WordNet* entailment relations

Thus, *T* entails *H* if a synset S_T including *T* is a hyponym of a synset S_H including *H*, considering transitivity. For example, *glucose* and *sugar* are lexically entailed because a path of an only hyponymy relation exists between a synset of *glucose* and a synset of *sugar*. Another example is given for the lexical units *kill* and *death*, where synsets containing them are related through a *WordNet* entailment relation.

3.3 Multiwords

There are many multiwords in *WordNet* showing useful semantic relations with other words and multiwords. The recognition of multiwords needs an extra processing in order to normalize their components. For example, the recognition of the multiword *came_down* requires the previous extraction of the lemma *come*, because the multiword present in *WordNet* is *come_down*.

The variation of multiwords does not happen only because of lemmatization. Sometimes there are some characters that change as, for example, a dot in an acronym or a proper noun with different wordings. For this reason, a fuzzy matching between candidate and *WordNet* multiwords was implemented using the edit distance of Levenshtein [8]. If the two strings differ in less than 10%, then the matching is permitted. For example, the multiword *Japanise_capital* in hypothesis 345 of the training corpus was translated into the *WordNet* multiword *Japanese_capital*, allowing the entailment between *Tokyo* and it. These are some other examples of entailment after multiword recognition; because of synonymy *blood_glucose* and *blood_sugar*, *Hamas* and *Islamic_Resistance_Movement* or *Armed_Islamic_Group* and *GIA* can be found; because of hyponymy, some examples in the corpus are: *war_crime* entails *crime* and *melanoma* entails *skin_cancer*.
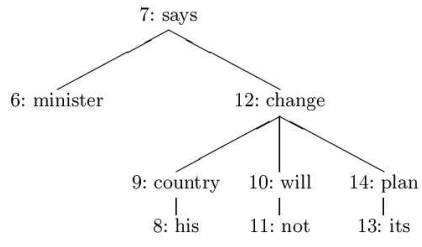
3.4 Negation and Antonymy

Negation is detected after finding leaves with a negation relationship with its father in the dependency tree. This negation relationship is then propagated to its ancestors until the head. For example, figures 2 and 3 show an excerpt of the dependency trees for the training examples 74 and 78 respectively. Negation at node 11 of text 74 is propagated to node 10 (*neg(will)*) and node 12 (*neg(change)*). Negation at node 6 of text 78 is propagated to node 5 (*neg(be)*). Therefore, entailment is not possible between a lexical unit and its negation. For example, before considering negation, node 5 in text 78 (*be*) entails node 4 in hypothesis 78 (*be*). Now, this entailment is not possible.

The entailment between nodes affected by negation is implemented considering the antonymy relation of *WordNet*, and applying the previous processing to them (sections 3.1, 3.2, 3.3). For example, since node 12 in text 74 is negated (*neg(change)*), the antonyms of *change* are considered in the entailment relations between text and hypothesis. Thus, *neg(change)* in text entails *continue* in the hypothesis because the antonym of *change*, *stay*, is a synonym of *continue*.
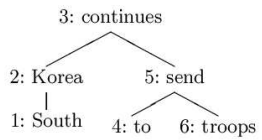
4 Mapping between Dependency Trees

Dependency trees give a structured representation for every text and hypothesis. The notion of mapping [13] between dependency trees can give an idea about how semantically similar are two text snippets; this is because a certain semantic information is implicitly contained into dependency trees. The technique used here to evaluate a matching between dependency trees is inspired in Lin's proposal [10] and is based on the notion of tree inclusion [6].

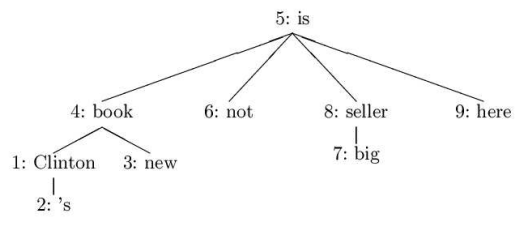**Text 74: ...minister says his country will not change its plan...**

```
                        7: says
                   ╱              ╲
          6: minister            12: change
                              ╱       |       ╲
                      9: country   10: will   14: plan
                          |           |          |
                        8: his     11: not     13: its
```

Hipothesis 74: South Korea continues to send troops

```
                    3: continues
                  ╱              ╲
          2: Korea              5: send
              |               ╱       ╲
          1: South        4: to      6: troops
```
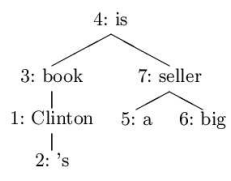
Fig. 2. Dependency trees for pair 74 from training corpus. Entailment is *TRUE*.


Text 78: Clinton's new book is not big seller here

```
                        5: is
                 ╱      |      |      ╲
          4: book   6: not  8: seller  9: here
          ╱     ╲              |
   1: Clinton  3: new       7: big
       |
     2: 's
```

Hypothesis 78: Clinton's book is a big seller

```
                    4: is
                  ╱        ╲
          3: book         7: seller
              |          ╱        ╲
        1: Clinton    5: a      6: big
              |
           2: 's
```

Fig. 3. Dependency trees for pair 78 from training corpus. Entailment is *FALSE*.

An abstract hypothesis' dependency tree and its respective abstract text's dependency tree are shown in figure 4, as an example. Thick lines are used to represent both the hypothesis' matching branches and the text's branches containing nodes that show a lexical entailment with a node from the hypothesis. Note that not every node from a branch of the text's dependency tree must show a lexical entailment with another node from the hypothesis, while a branch from the hypothesis is considered a "matching branch" only if all its nodes are involved in a lexical entailment with a node from the respective branch from the text's dependency tree.
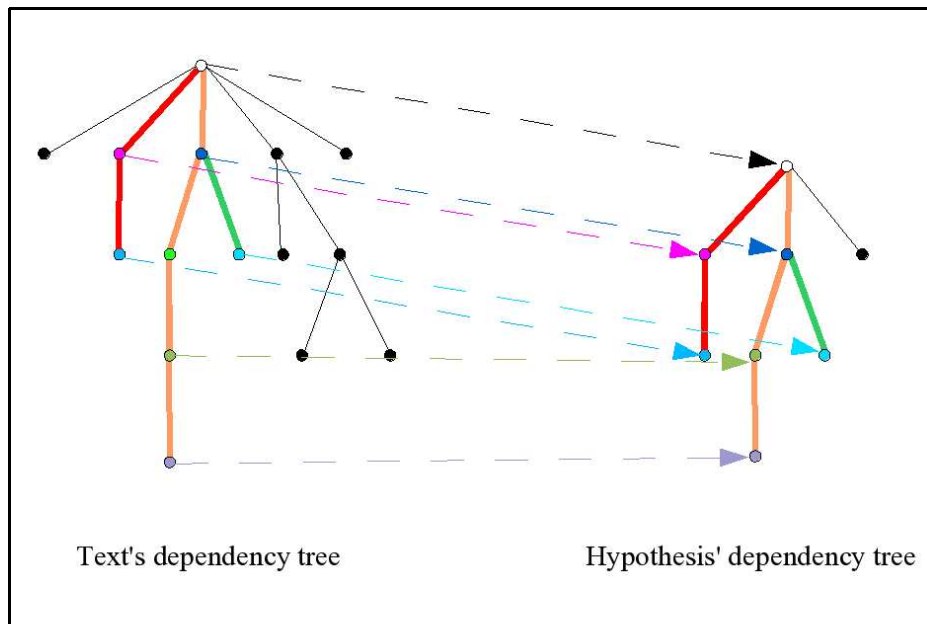
Fig. 4. Example for hypothesis' matching branches.

The subtree conformed by all the matching branches from a hypothesis' dependency tree is included in the respective text's dependency tree. The work hypothesis assumes that the larger is the included subtree of the hypothesis' dependency tree, the more semantically similar are the text and the hypothesis. Thus, the existence or absence of an entailment relation from a text to its respective hypothesis is determined by means of the portion of the hypothesis' tree that is included in the text's tree.

Informally, this *tree overlap* measures how large is the hypothesis' dependency subtree *included* in the text's dependency tree with respect to the whole hypothesis' dependency tree. A higher degree of matching between dependency trees has been taken as indicative of a semantic relation. The threshold to determine whether there exists an entailment relation between a text and a hypothesis is obtained after training the system with the development corpus.

5 Experiments

Some experiments were accomplished in order to obtain feedback about successive improvements made to our system. For this purpose, several settings were trained over the development corpus and evaluated against the test corpus.

- System 1
  - Lexical level: No special processing for lexical entailment, but the coincidence between a word from the text and the hypothesis.
  - Entailment decision: build a decision tree using C4.5 [11] over the training corpus and use this tree to classify the test samples. The set of attributes for building the decision tree were:
    * Number of nodes in the hypothesis' dependency tree.
    * Number of nodes in the hypothesis' dependency tree not entailed by any node in the text's dependency tree.
    * Percentage of entailed nodes from the hypothesis' dependency tree.
- System 2
  - Lexical level: lexical entailment as described in section 3.
  - Entailment decision: same as system 1.
- System 3
  - Lexical level: same as system 2.
  - Entailment decision: same as systems 1 and 2, but adding boolean attributes to the decision tree specifying whether nodes showing a subject or object relations with their fathers have failed or not (i.e., if they have not been entailed by any node from the text).
- System 4
  - Lexical level: same as systems 2 and 3.
  - Entailment decision: applying the algorithm from section 4 based on the notion of tree inclusion [6].

6 Results

Overall results are shown in table 1. The behavior of all the systems is quite similar except for system 4 that obtains the lower accuracy. The use of the lexical entailment module based on *WordNet* slightly increases accuracy (system 2 with respect to system 1); however, the inclusion of attributes in the decision tree related to the syntactic role (subject and object) does not improve the performance in our setting (system 3). Finally, the overlapping algorithm based on the notion of tree inclusion did not obtain the expected performance (system 4).

Some questions arise about the mapping between dependency trees approach. Though the notion of inclusion is not enough for RTE, some other notions such as tree alignment distance [2] [4] or tree edit distance [2] [4]] seem promising as shown in [7]. Nevertheless, the results obtained by systems 2 and 3 are close to those obtained with the best approaches in PASCAL RTE Challenge [3].

Table 1. Accuracy values of the systems

	Accuracy
System 1	55.87%
System 2	56.37%
System 3	56.25%
System 4	54.75%

7 Conclusions and Future Work

The use of lexical resources such as *WordNet* aimed at recognizing entailment and equivalence relations at the lexical level for improving system's performance. In this direction, the next step is to recognize and evaluate entailment between numeric expressions, Named Entities and temporal expressions.

A mapping of dependency trees based on the notion of inclusion (as shown here) is not enough to tackle appropriately the problem, with the possible exception of Comparable Document [3] tasks. A higher lexical overlap does not mean a semantic entailment and a lower lexical overlap does not mean different semantics. Other mapping approaches based on the notions of tree edit distance or tree alignment distance seem more promising [7].

Both lexical and syntactic issues to be improved have been detected. At the lexical level, some kind of paraphrasing detection would be useful; for example, in pair 96 of the training corpus (see table 2) is necessary to detect the equivalence between *same-sex* and *gay* or *lesbian*; or, in pair 128 (see table 2), *come into conflict with* and *attacks* must be detected as equivalent. Previous work has been developed; for example, Szpektor et al. (2004) [12] propose a web-based method to acquire entailment relations; Barzilay and Lee (2003) [1] use multiple-sentence alignment to learn paraphrases in an unsupervised way; or Hermjakob et al. (2002) [5] show how *WordNet* can be extended as a reformulation resource.

Table 2. Pairs 96 and 128 from the training corpus

Text 96: The Massachusetts Supreme Judicial Court has cleared the way for lesbian and gay couples in the state to marry, ruling that government attorneys "failed to identify any constitutionally adequate reason" to deny them the right.
Hypothesis 96: U.S. Supreme Court in favor of same-sex marriage
Text 128: Hippos do come into conflict with people quite often.
Hypothesis 128: Hippopotamus attacks human.

Sometimes, two related words are not considered because their lemmas (provided by the dependency parser) are different or a semantic relation between them can not be

found; for example, in pair 128 of the training corpus the relations between *Hippos* and *Hippopotamus* and the relation between *people* and *human* are not detected.

Other problem is that, in certain cases, a high matching between hypothesis' nodes and text's nodes is given but, simultaneously, hypothesis' branches match with disperse text's branches; then, syntactic relations between substructures of the text and the hypothesis must be analyzed in order to determine the existence of an entailment.

Some other future lines of work include:

– A detailed analysis of the corpora, with the aim of determining what kinds of inference are necessary in order to tackle successfully the entailment detection. For example: temporal relations, spatial relations, numeric relations, relations between named entities, paraphrase detection, etcetera; and the development of the corresponding subsystems.
– The development of improved mapping algorithms between trees, such as the tree edit distance or an alignment distance [2] [4].

Hence, it is observed that for RTE is necessary to tackle a wide set of linguistic phenomena in a specific way, both at the lexical level and at the syntactic level.

Acknowledgments

References

1. R. Barzilay and L. Lee. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In *NAACL-HLT*, 2003.
2. P. Bille. Tree Edit Distance, Alignment Distance and Inclusion. Technical Report TR-2003-23, IT Technical Report Series, March 2003.
3. I. Dagan, O. Glickman, and B. Magnini. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, Southampton, UK*, pages 1–8, April 2005.
4. R. Gusfield. *Algoritms on Strings, Trees and Sequences*. Cambridge University Press, 1997.
5. U. Hermjakob, A. Echibabi, and D. Marcu. Natural Language Based Reformulation Resource and Web Exploitation for Question Answering. In *Proceedings of TREC*, 2002.
6. P. Kilpeläinen. Tree Matching Problems with Applications to Structured Text Databases. Technical Report A-1992-6, Department of Computer Science, University of Helsinki, Helsinki, Finland, November 1992.
7. M. Kouylekov and B. Magnini. Recognizing Textual Entailment with Tree Edit Distance Algorithms. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, Southampton, UK*, pages 17–20, April 2005.
8. V. I. Levensthein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. In *Soviet Physics - Doklady*, volume 10, pages 707–710, 1966.
9. D. Lin. Dependency-based Evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems, Granada, Spain*, May 1998.

10. D. Lin and P. Pantel. DIRT - Discovery of Inference Rules from Text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 323–328, 2001.

11. J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, 1993.

12. I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. Scaling Web-Based Acquisition of Entailment Relations. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP-04)*, 2004.

13. G. Valiente. An Efficient Bottom-Up Distance Between Trees. In *Proceedings of the International Symposium on String Processing and Information REtrieval, SPIRE*, pages 212–219, 2001.