Combining Lexical-Syntactic Information with Machine Learning for Recognizing Textual Entailment

Arturo Montejo-Ráez, Jose Manuel Perea, Fernando Martínez-Santiago, Miguel Ángel García-Cumbreras, Maite Martín-Valdivia, Alfonso Ureña-López
Dpto. de Informática, Universidad de Jaén
Campus de las Lagunillas s/n, 23071 - Jaén
{amontejo, jmperea, dofer, magc, maite, laurena}@ujaen.es

Abstract

This document contains the description of the experiments carried out by SINAI group. We have developed an approach based on several lexical and syntactic measures integrated by means of different machine learning models. More precisely, we have evaluated three features based on lexical similarity and 11 features based on syntactic tree comparison. In spite of the relatively straightforward approach we have obtained more than 60% for accuracy. Since this is our first participation we think we have reached a good result.

1 Approach description

We fill face the textual entailment recognition using Machine Learning methods, i.e. identifying features that characterize the relation between hypothesis and associated text and generating a model using existing entailment judgements that will allow us to provide a new entailment judgement agains unseen pairs text-hypothesis. This approach can be split into the two processes shown in Figures 1 and 2.

In a more formal way, given a text *t* and an hypothesis *h* we want to define a function *e* which takes these two elements as arguments and returns and answer to the entailment question:

$$e(t,h) = \begin{cases} YES & \text{if } h \text{ is entailed by } t \\ NO & \text{otherwise} \end{cases} \quad (1)$$

Now the question is to find that ideal function $e(t,h)$. We will approximate this function using a binary classifier:

Figure 1: Training processes

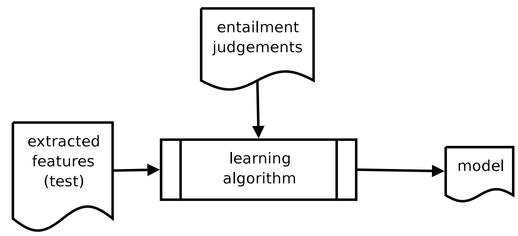
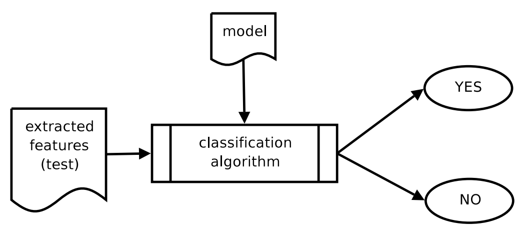


Figure 2: Classification processes



$$\hat{e}(t,h) = bc(f,m) \quad (2)$$

where
bc is a binary classifier
f is a set of *features*
m is the learned model for the classifier

Therefore, it only remains to select a binary classifier and a feature extraction method. We have performed two experiments with different choices for both decisions. These two experiments are detailed below.

Proceedings of the Workshop on Textual Entailment and Paraphrasing, pages 78–82,
Prague, June 2007.

1.1 Lexical similarity

This experiment approaches the textual entailment task being based on the extraction of a set of lexical measures that show the existing similarity between the hypothesis-text pairs. Our approach is similar to (Ferrandez et al., 2007) but we make matching between similar words too while (Ferrandez et al., 2007) apply exact matching (see below).

The first step previous to the calculation of the different measures is to preprocess the pairs using the English *stopwords* list. Next we have used the GATE[1] architecture to obtain the stems of tokens. Once obtained stems, we have applied four different measures or techniques:

- **Simple Matching**: this technique consists of calculating the semantic distance between each stem of the hypothesis and text. If this distance exceeds a threshold, both stems are considered similar and the similarity weight value increases in one. The accumulated weight is normalized dividing it by the number of elements of the hypothesis. In this experiment we have considered the threshold 0.5. The values of semantic distance measure range from 0 to 1. In order to calculate the semantic distance between two tokens (stems), we have tried several measures based on WordNet (Alexander Budanitsky and Graeme Hirst, 2001). **Lin's similarity measure** (Lin, 1998) was shown to be best overall measures. It uses the notion of information content and the same elements as Jiang and Conrath's approach (Jiang and Conrath, 1997) but in a different fashion:

$$sim_L(c_1, c_2) = \frac{2 \times \log p(lso(c_1, c_2))}{\log p(c_1) + \log p(c_2)}$$

where c_1 and c_2 are synsets, *lso*(c_1,c_2) is the information content of their lowest superordinate (most specific common subsumer) and *p(c)* is the probability of encountering an instance of a synset *c* in some specific corpus (Resnik, 1995). The Simple Matching technique is defined in the following equation:

$$SIM_{matching} = \frac{\sum_{i \in H} similarity(i)}{|H|}$$

where *H* is the set that contains the elements of the hypothesis and *similarity(i)* is defined like:

$$similarity(i) = \begin{cases} 1 & \text{if } \exists j \in T sim_L(i,j) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

- **Binary Matching**: this measure is the same that the previous one but modifying the *similarity* function:

$$similarity(i) = \begin{cases} 1 & \text{if } \exists j \in T \; i = j \\ 0 & \text{otherwise} \end{cases}$$

- **Consecutive Subsequence Matching**: this technique relies on forming subsequences of consecutive stems in the hypothesis and matching them in the text. The minimal size of the consecutive subsequences is two and the maximum is the maximum size of the hypothesis. Every correct matching increases in one the final weight. The sum of the obtained weights of the matching between subsequences of a certain size or length is normalized by the number of sets of consecutive subsequences of the hypothesis created for this length. These weights are accumulated and normalized by the size of the hypothesis less one. The Consecutive Subsequence Matching technique is defined in the following equations:

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

where SH_i is the set that contains the subsequences of the hypothesis with *i* size or length and *f*(SH_i) is defined like:

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

where

$$matching(i) = \begin{cases} 1 & \text{if } \exists k \in ST_i \; k = j \\ 0 & \text{otherwise} \end{cases}$$

where ST_i represents the set that contains the subsequences with *i* size from text.

- **Trigrams**: this technique relies on forming trigrams of words in the hypothesis and matching them in the text. A trigram is a group of

[1] http://gate.ac.uk/

three words. If a hypothesis trigram matches in text, then the similarity weight value increases in one. The accumulated weight is normalized dividing it by the number of trigrams of the hypothesis.

1.2 Syntactic tree comparison

Some features have been extracted from pairs hypothesis-text related to the syntactic information that some parser can produce. The rationale behind it consists in measuring the similarity between the syntactic trees of both hypothesis and associated text. To do that, terms appearing in both trees are identified (we call this *alignment*) and then, graph distances (number of nodes) between those terms in both trees are compared, producing certain values as result.

In our experiments, we have applied the COLLINS (Collins, 1999) parser to generate the syntactic tree of both pieces of text. In Figure 3 the output of the syntactic parsing for a sample pair is shown. This data is the result of the syntactical analysis performed by the mentioned parser. A graph based view of the tree corresponding to the hypothesis is drawn in Figure 4. This graph will help us to understand how certain similarity measures are obtained.

Figure 3: Syntactic trees of sample hypothesis and its associated text

```
<t>
(TOP (S (LST (LS 0302) (. .)) (NP (JJ Next) (NN year))
(VP (VBZ is) (NP (NP (DT the) (JJ 50th) (NN anniversary))
(PP (IN of) (NP (NP (DT the) (NNP Normandy) (NN invasion)
(, ,)) (NP (NP (DT an)(NN event)) (SBAR (IN that) (S (VP
(MD would) (RB n't) (VP (VB have) (VP (VBN been) (ADJP
(JJ possible)) (PP (IN without) (NP (NP (DT the) (NNP
Liberty) (NN ships.)) (SBAR (S (NP (DT The) (NNS
volunteers)) (VP (VBP hope) (S (VP (TO to) (VP (VB raise)
(NP (JJ enough) (NN money)) (S (VP (TO to) (VP (VB sail)
(NP (DT the) (NNP O'Brien)) (PP (TO to) (NP (NNP France)))
(PP (IN for)(NP (DT the) (JJ big) (NNP D-Day) (NN celebration)
(. .)))))))))))))))))))))))))
</t>

<h>
(TOP (S (NP (NP (CD 50th) (NNP Anniversary)) (PP (IN of)
(NP (NNP Normandy) (NNP Landings)))) (VP (VBZ lasts) (NP
(DT a) (NN year) (. .)))))
</h>
```

From the sample above, the terms *normandy*, *year* and *anniversary* appear in both pieces of text. We say that these terms are “aligned”. Therefore, for the three possible pairs of aligned terms we can compute the distance, in nodes, to go from one term to the other at each tree. Then, the difference of these distances is computed and some statistics are generated. We can summarize the process of computing this differences in the algorithm detailed in Figure 6.

Figure 4: Syntact tree of sample hypothesis

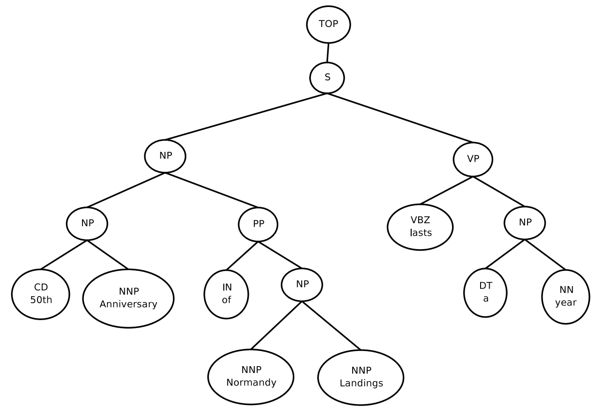
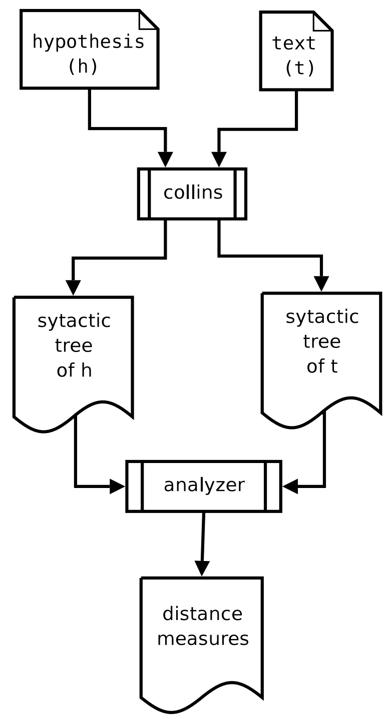


Figure 5: Tree comparison process



For instance, in the tree represented in Figure 4 we can see that we have to perform 5 steps to go from node *Anniversary* to node *Normandy*. Since there are no more possible occurrences of these two terms, then the minimal distance between them is 5. This value is also measured on the tree corresponding to the text, and the absolute difference between these two minimal distances is stored in order to compute final feature weights consisting in basic statistical values. The algorithm to obtain the distribution of distance differences is detailed in Figure 6.

Figure 6: Extraction of features based on syntactic distance

Input:
  a syntactic tree of the hypothesis S_h
  a syntactic tree of the text S_t
Output :
  the set of distance differences
  $Dd = \{dd_{ij} : t_i, t_j \in T\}$
Pseudo code:
  $T \leftarrow$ *aligned terms between* S_h *and* S_t
  $Dd \leftarrow \emptyset$
  for $i = 1..n$ do
    for $j = i + 1..n$ do
      $dist_h \leftarrow$ *minimal distance between nodes* t_i and t_j in S_h
      $dist_t \leftarrow$ *minimal distance between nodes* t_i and t_j in S_t
      $dd_{ij} \leftarrow |dist_h - dist_t|$
      $Dd \leftarrow \{dd_{ij}\} \cup Dd$
    end-for
  end-for

The statistics generated from the resulting list of distances differences Dd are the following:

1. The number of aligned terms (3 in the given example).
2. The number of matched POS values of aligned terms, that is, if the term appears with the same POS label in both texts (in the example *Anniversary* differs in the POS label assigned).
3. The number of unmatched POS labels of aligned terms.
4. The average distance in nodes through the syntactic tree to go from one aligned term to another.
5. The minimal distance difference found.
6. The maximal distance difference found.
7. The standard deviation of distance differences.

Table 1: Results with TiMBL and BBR classifiers (*Exp5* is the only official result reported in this paper).

Experiment	Classifier	Accuracy
Exp1	BBR	0.6475
Exp2	BBR	0.64625
Exp3	BBR	0.63875
Exp4	TiMBL	0.6062
Exp5	TiMBL	0.6037
Exp6	TiMBL	0.57

In a similar way, differences in the depth level of nodes for aligned terms are also calculated. From the example exposed the following values were computed:

```
* Aligned         3
* MatchedPOS      2
* UnmatchedPOS    1
* AvgDistDiff     0.0392156863
* MinDistDiff     0.0000000000
* MaxDistDiff     0.0588235294
* StdevDistDiff   0.0277296777
* AvgDepthDiff    2.0000000000
* MinDepthDiff    1.0000000000
* MaxDepthDiff    3.0000000000
* StdevDepthDiff  0.8164965809
```

2 Experiments and results

The algorithms used as binary classifiers are two: *Bayesian Logistic Regression* (BBR)[2] and TiMBL (Daelemans et al., 1998). Both algorithms have been trained with the *devel* data provided by the organization of the Pascal challange. As has been explained in previous sections, a model is generated via the supervised learning process. This model *m* is then feed into the classification variant of the algorithm, which will decide whether a new hypothesis sample is entailed by the given text or not.

The experiments and results are shown in Table 1: where:

- **Exp1** uses four features: three lexical similarities ($SIM_{matching}$ + $CSS_{matching}$ + Trigrams) and Syntactic tree comparison.

[2] http://www.stat.rutgers.edu/~madigan/BBR/ [available at March 27, 2007]

- **Exp2** uses five features: four lexical similarities ($SIM_{matching}$ + $CSS_{matching}$ + Trigrams + $BIN_{matching}$) and Syntactic tree comparison.
- **Exp3** uses only three lexical similarities ($SIM_{matching}$ + $CSS_{matching}$ + Trigrams).
- **Exp4** uses the four lexical similarities ($SIM_{matching}$ + $CSS_{matching}$ + Trigrams + $BIN_{matching}$)
- **Exp5** uses only three lexical similarities ($SIM_{matching}$ + $CSS_{matching}$ + Trigrams).
- **Exp6** uses four features: three lexical similarities ($SIM_{matching}$ + $CSS_{matching}$ + Trigrams) and Syntactic tree comparison.

As we expected, the best result we have obtained is by means of the integration of the whole of the features available. More surprising is the good result obtained by using lexical features only, even better than experiments based on syntactical features only. On the other hand, we expected that the integration of both sort of features improve significatively the performance of the system, but the improvement respect of lexical features is poor (less than 2%). . Similar topics share similar vocabulary, but not similar syntax at all. Thus, we think we should to investigate semantic features better than the syntactical ones.

3 Conclusions and future work

In spite of the simplicity of the approach, we have obtained remarkable results: each set of features has reported to provide relevant information concerning to the entailment judgement determination. On the other hand, these two approaches can be merged into one single system by using different features all together and feeding with them several binary classifiers that could compose a voting system. We will do that combining TiMBL, SVM and BBR.We expect to improve the performance of the entailment recognizer by this integration.

Finally, we want to implement a hierarchical architecture based on constraint satisfaction networks. The constraints will be given by the set of available features and the maintenance of the integration across the semantic interpretation process.

4 Acknowledgements

This work has been partially financed by the TIMOM project (TIN2006-15265-C06-03) granted by the Spanish Government Ministry of Science and Technology and the RFC/PP2006/Id_514 granted by the University of Jaén.

References

Alexander Budanitsky and Graeme Hirst. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 1998. Timbl: Tilburg memory based learner, version 1.0, reference guide.

Oscar Ferrandez, Daniel Micolo, Rafael Mu noz, and Manuel Palomar. 2007. Técnicas léxico-sintácticas para reconocimiento de inmplicación textual. . *Tecnologías de la Informacón Multilingüe y Multimodal. In press.*

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*, Taiwan.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*.

Philip Resnik. 1995. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal.