Combining Specialized Entailment Engines for RTE-4

Elena Cabrio[1,2]**, Milen Kouylekov**[1] **and Bernardo Magnini**[1]
FBK-Irst[1]
University of Trento[2]
38050, Povo, Trento, Italy
{cabrio, kouylekov, magnini}@fbk.eu

Abstract

The main goal of FBK-irst participation at RTE-4 was to experiment the use of combined specialized entailment engines, each addressing a specific phenomena relevant to entailment. The approach is motivated since textual entailment is due to the combination of several linguistic phenomena which interact among them in a quite complex way. We were driven by the following two considerations: (i) devise a general framework, based on distance between T and H, flexible enough to allow the combination of single entailment engines; (ii) provide a modular approach through which evaluate progresses on single aspects of entailment, using specialised traning and test dataset. For RTE-4 we used two simple entailment engines, one addressing negation and the other lexical similarity, with a linear combination of their respective distances on T-H pairs.

1 Introduction

Textual entailment, the problem of establishing entailment relations between a pair of textual fragments, requires the intervention of knowledge at different levels. Several studies in the literature ((Clark, 2007), (Vanderwende, 2006)) point out that the lexical, syntactic and world knowledge levels can be analyzed and exploited in order to fully identify and recognize the entailment between a *text* (T) and a *hypothesis* (H). This is, at least partially, reflected by the RTE datasets, where the existence of linguistic properties characterizing the relation of entailment between the provided text snippets come to light. However, the problem of how such aspects interact with respect to entailment has not been fully investigated so far.

As an example, consider the following pair from the RTE-4:

```
<pair id="71" entailment="CONTRADICTION"
task="IR">
<t>Child welfare workers were struggling
Thursday to find foster care facilities
for some 437 children removed from a
polygamist sect in Texas amid allegations
of widespread sexual abuse. The children
will not be returned to their families.
</t>
<h>US sect children are sent home.</h>
</pair>

<pair id="400" entailment="ENTAILMENT"
task="QA">
<t>The polygraph came along in 1921,
invented by John A. Larson, a University
of California medical student working with
help from a police official. The device
ostensibly detects when a person is lying
by monitoring and recording certain body
changes affected by a person's emotional
condition.</t>
<h>The polygraph is a device that
ostensibly detects when a person is not
telling the truth.</h></pair>
```

As can be noticed, multiple linguistic aspects are relevant for entailment. For instance, in pair 71, to know that *be returned* and *sent* are synonyms and that one of the two verbs is the scope of the negation is fundamental to state that T does not entail H. Similarly, in pair 400, to know that *lying* is antonym of

telling the truth, and that the antonym is the scope of the negation allows to state that there is entailment between T and H.

The main goal of FBK-irst participation at RTE-4 was to experiment the use of combined specialized entailment engines, each addressing a specific phenomena relevant to entailment. We were driven by the following two considerations: (i) devise a general framework, based on distance between T and H, flexible enough to allow the combination of single entailment engines; (ii) provide a modular approach through which evaluate progresses on single aspects of entailment, using specialised traning and test dataset. For RTE-4 we used two simple entailment engines, one addressing negation and the other lexical similarity, with a linear combination of their respective distances on T-H pairs.

The paper is structured as follows. Section 2 introduces our approach to RTE, and provides details on the architecture of the system. Section 3 and Section 4 describe the specialized entailment engine we have implemented so far, while Section 5 discusses the results we have obtained in the RTE4 Challenge. Section 6 concludes the paper drawing final remarks and presents some directions for future works.

2 System overview

The EDITS system (Edit Distance Textual Entailment Suite) assumes that the distance between *text* and *hypothesis* is a characteristic that separates the positive pairs, for which entailment holds, from the negative pairs, for which entailment does not hold (see also (Kouylekov and Magnini, 2006)). Such distance is computed as the cost of the editing operations (*i.e.* insertion, deletion and substitution) which are required to transform the text T into the hypothesis H. Each edit operation on two text fragments A and B (denoted as $A \rightarrow B$) has an associated cost (denoted as $\gamma(A \rightarrow B)$). The entailment score for a *text-hypothesis* pair is calculated on the minimal set of edit operations that transform T into H. An entailment relation is assigned to a T-H pair only if the overall cost of the transformations is below a certain threshold empirically estimated over training data.

2.1 System architecture

The idea underlying our approach is to have different independent entailment engines, each of which able to deal with an aspect of the language variability (*e.g.* negation, modals). In every engine, the cost of edit operations should be defined according to the specific linguistic phenomenon it should cope with. The cost schemes of the different specialized engines are designed in order not to intersect. If the costs of the edit operations are set as not 0 for a certain phenomena, they are set as 0 for the aspects that are considered by another module.

The output of the whole system is therefore defined by the sum of the edit distances produced by each module, as showed in Figure 1.

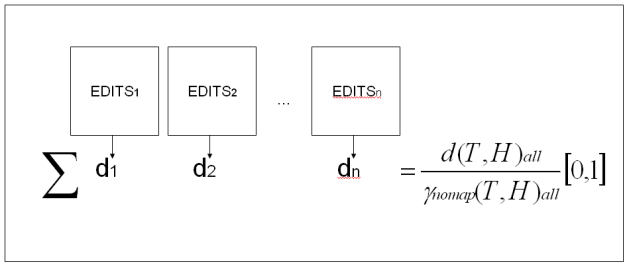

Figure 1: EDITS System.

In the entailment score function reported in Figure 1, the sum of the distances between T and H provided by each module $d(T,H)_{all}$ is divided for $\gamma_{nomap}(T,H)_{all}$, i.e. the sum of the *no mapping* distances equivalent to the cost of inserting the entire text of H, and deleting the entire text of T. The entailment score function has a range from 0 (when T is identical to H), to 1 (when T is completely different from H).

As introduced before, although the different linguistic phenomena can be dependent on each other in different and complex ways, for the time being we decided to sum the specialized modules, postponing to future work the open issue of how to combine them according to the dependencies.

Each EDITS is composed by the following modules, showed in Figure 2: (i) a distance algorithm, which determines the best (less costly) sequence of edit operations (insertion, deletion and substitution) that allow to transform T into H; (ii) a cost schema, that determines the cost of the three edit operations,

and (iii) a set of entailment rules, each with a probability associated representing the degree of confidence of the rule. In EDITS, entailment rules can be at different levels (e.g. lexical, syntactic, etc.) and can be either generated from existing resources (e.g. WordNet) or manually defined.

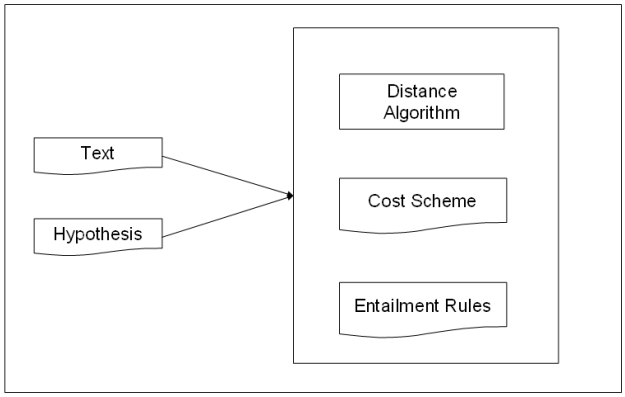

Figure 2: EDITS architecture.

2.2 Algorithms

For RTE4, two distance-based algorithms have been used: Linear Distance and Tree Edit Distance.

Linear Distance (LD) As for Linear Distance, *Levenshtein Distance* has been applied to RTE (Kouylekov and Magnini, 2006), by converting both T and H into sequences of words. Accordingly, edit operations have been defined as follows:

- **Insertion** ($\Lambda \rightarrow A$): insert a word A from hypothesis into text.
- **Deletion** ($A \rightarrow \Lambda$): delete a word A from T.
- **Substitution** ($A \rightarrow B$): substitute a word A from T with a word B from H.

Tree Edit Distance (TED) As regards Tree Edit Distance, we have proposed an implementation for RTE based on (Zhang and Shasha, 1990), where the dependency trees of both T and H are considered (Kouylekov and Magnini, 2006). Edit operations are defined in the following way:

- **Insertion** ($\Lambda \rightarrow A$): insert a node A 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** ($A \rightarrow \Lambda$): delete a node A from the dependency tree of T. When A is deleted all its children are attached to the parent of A. It is not required to explicitly delete the children of A, as they are going to be either deleted or substituted in a following step.
- **Substitution** ($A \rightarrow B$): change the label of a node A in the source tree into a label of a node B of the target tree. In case of substitution the relation attached to the substituted node is changed with the relation of the new node.

2.3 Combining specialized Cost Schemes for Edit Operations

The core of the edit distance approach is the mechanism for the definition of the cost of edit operations. This mechanism is defined separately from the distance algorithm and should reflect the knowledge of the user about the processed data. The principle behind it is to capture certain phenomena that facilitate the algorithm to assign small distances to positive T-H pairs, and large distances to negative pairs. Different semantic representations of the text allow different ways of defining the cost of edit operations. In the following sections we describe the EDITS engines implemented so far.

3 EDITSneg

As introduced before, the model is composed by the sum of different independent EDITS modules, each of which should be able to deal with a specific phenomena of the language variability. The first module that has been implemented works on the negative polarity items (NPIs) issue.

Some of the systems presented in the previous editions of the RTE challenges attempted specific strategies to afford this matter. For instance, (Snow, 2006) presents a framework for recognizing textual entailment that focuses on the use of syntactic heuristics to recognize false entailment. Among the others, heuristics concerning negation mismatch and antonym match are defined. More recently, in (Tatu, 2007) the logic representation of sentences with negated concepts was altered to mark as negated the entire scope of the negation. In particular, (Harabagiu, 2006) focuses on contradictions that originate when using negation, antonymy, and

semantic and pragmatic information associated with the *contrast* discourse relation.

In conformity with the system architecture described in Section 2.1, EDITSneg is composed by a cost schema that sets specific costs for edit operations concerning negation (as shown in Example 1), and uses the Linear Distance algorithm (Levenshtein distance calculated on tokens) to determine the less costly sequence of edit operations that allow to transform T into H.

Example 1: Rule of the cost schema (for insertion).

```
<!--Insertion of a token (that is preceded
by a negation) from H  in T -->
<rule name="insertion_negatedword">
<right><syntax><token><text>\$VAR{A}</text>
<attribute name="HasNegation">TRUE
</attribute></token></syntax></right>
<score>30</score>
</rule>
```

The underlying intuition is that assigning high costs to edit operations that involve negative polarity items should prevent the system to assign positive entailment to a T-H pair in which one of the two fragments contradicts (is negated by) the other. For this reason, the evaluation measure rewarded is *precision no*. In this schema, all other words but the negative polarity items have a zero cost of insertion, deletion and substitution.

We process negation focusing on direct licensors of negation such as overt negative markers (*not*, and the bound morpheme *n't*), negative quantifiers (*no*, *nothing*), strong negative adverbs (*never*). Also contradictions arising from the use of antonyms are taken into consideration. The preprocessing module annotates the output of TextPro (Pianta, 2008) with other linguistic information: detection of negated words (the attribute *"HasNegation"* is added to the words that are preceded by a negation) and antonymy information derived from WordNet (Fellbaum, 1998) (the attribute *"HasAntonym"* is added to the words in T that has an antonym in H and viceversa).

3.1 Experiments on an *ad hoc* dataset for negation

In order to make experiments on the negation phenomena, an artificial balanced dataset has been built. It is composed of 66 T-H pairs (32 entailment *yes*; 34 entailment *no*). The words of the two fragments T and H are aligned and there are no differences between T and H except for the presence of direct licensors of negation or antonyms (both in T and H; only in T and not in H or viceversa; neither in T nor in H). This criterion has been chosen in order to avoid other linguistic phenomena influencing the decisions of the system. The text snippets T have been extracted shortening or simplifying T or H fragments of previous versions of RTE datasets, and snippets H have been added according to the defined criteria, as shown in Example 2.

Example 2: Negation dataset.

```
<pair id="1" entailment="NO">
<t>ECB spokeswoman, Regina Schueller
worked for Italy's La Repubblica
newspaper.</t>
<h>ECB spokeswoman, Regina Schueller
never worked for Italy's La Repubblica
newspaper.</h>

<pair id="7" entailment="NO">
<t>Sudan refused to allow U.N. troops
in Darfur.</t>
<h>Sudan accepted to allow U.N. troops
in Darfur.</h>
</pair>
```

Data concerning the results of EDITSneg on the negation dataset are reported in Table 1.

		EDITS_neg
dataset-neg	**Precision**	0.710
	Recall	0.870
	F-measure	0.782
	Accuracy	**0.772**

Table 1: *Results on the negation dataset.*

In most of the cases, the system fails when antonymy information extracted from WordNet is incomplete, and when a consideration of the syntactic structure of the sentence is unavoidable to be able to assign the correct entailment relation (e.g. *No other details were available* entails *Other details were not available*).

To improve the precision, it would be worth processing the dataset with a parser and exploiting syntactic information for the annotation of the scope of negation in the individual sentences (this would also reduce the number of the *false negative* produced by the system, as discussed in Section 5).

4 EDITSlex

The second module that we are going to implement deals with lexical similarity. EDITSlex is composed by: i) a cost schema that sets specific costs for edit operations considering WordNet similarities among words; ii) the Linear Distance algorithm (Levenshtein distance calculated on tokens); iii) a set of entailment rules (with a probability associated representing the degree of confidence of each rule) generated exploiting the WordNet similarity (Pedersen, 2004) between the tokens in T and H, as described below.

For RTE4, EDITlex is not yet set as an independent module, but it is integrated in a more general system that considers all but the negation phenomena (EDITSall-but-neg), plus WordNet similarities. EDITSall-but-neg computes the costs of the edit operations basing on the following cost scheme:

$$\gamma(\Lambda \rightarrow A) = length(T)$$
$$\gamma(A \rightarrow \Lambda) = length(H)$$
$$\gamma(A,B) = \begin{cases} 0 & A = B \\ \gamma_{i+d}(A \rightarrow B) * (1 - p_{A \rightarrow B}) & A \rightarrow B \\ \gamma_{i+d}(A \rightarrow B) & otherwise \end{cases}$$

The cost of the insertion of a text fragment from H in T is equal to the length (*i.e.* the number of words) of T, and the deletion of a text fragment from T is equal to the length of H. The substitution cost is set to the sum of the insertion and the deletion of the text fragments, if they are not equal. This means that the algorithm would prefer to delete and insert text fragments rather than substituting them, in case they are not equal[1]. Setting the insertion and deletion costs respectively to the length of T and H is motivated by the fact that a shorter text T should not be preferred over a longer one T' while computing their overall mapping costs with the hypothesis H. Setting the costs to fixed values would in fact penalize longer texts (due to the larger amount of deletions needed) even though they are very similar to H.

As introduced before, in this system we estimate the cost of substitutions using WordNet similarity (Pedersen, 2004), a method that provide a quantitative measure of the degree to which two word senses are related. Among the different measures of relatedness that have been implemented in this software package, we choose the Adapted Lesk (Extended Gloss Overlaps). It works by finding overlaps in the glosses of the two synsets: the relatedness score is the sum of the squares of the overlap lengths.

$$\text{sub}_{Lesk}(A \rightarrow B) = \begin{cases} 0 & sim < 100 \\ \frac{sim}{500} & 100 < sim < 500 \\ 1 & sim > 500 \end{cases}$$

where *sim* is the value of the lexical similarity between A and B.

5 Results on RTE4 dataset and discussion

Our official results at RTE-4 Challenge are shown in Table 2. We submitted three runs for the two-way RTE task: the first one with EDITSneg, the second one with a combined system (EDITSneg+ EDITSallbutneg) and the third one with our standard system.

		First run	Second run	Third run
RTE4	**Acc**	0.54	0.546	0.57
	Avg pr.	0.4946	0.5516	0.553

Table 2: *Results on RTE 4*

Concerning the first two runs, we participated in the RTE challenge as a way to understand what our specialized modules could do with respect to more general systems of textual entailment.

Figure 3 shows the behaviour of EDITSneg on the RTE4 dataset.

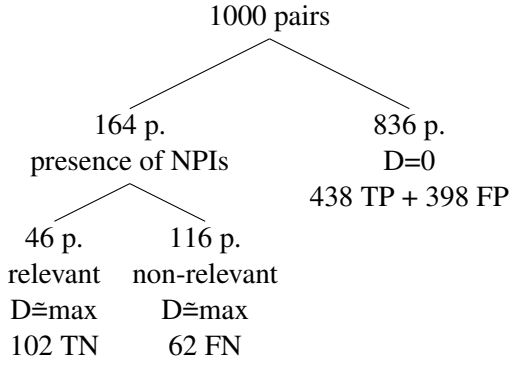

Figure 3: *Behaviour of EDITSneg on RTE4 dataset.*

As can be seen, we answered very few of the questions: only 164 of the possible 1000 pairs contained

[1]This remains valid for all the costs schemes.

negative polarity items, and the system assigned a *no entailment* answer. Among these, only for 64 pairs the detection of direct licensors of negation or of antonyms was crucial for a correct answer, as shown in Example 3.

Example 3: Negation in RTE 4 dataset.

```
<pair id="107" entailment="CONTRADICTION"
 task="IR">
<t>Giles Chichester's position was viewed
as untenable partly because he had been
given the job of a sleazebuster by Mr
Cameron to ensure the integrity of Tory MEP
expenses. He is not the leader of the Tory
MEPs.</t>
<h>Giles Chichester is the leader of the
Tories MEPs.</h></pair>

<pair id="167" entailment="CONTRADICTION"
task="IR">
<t>R. Kelly was acquitted of child porno-
graphy after the star witness Van Allen was
discredited after admitting she once stole
Kelly's \$20,000 diamond-studded watch from
a hotel.</t>
<h>R. Kelly was convicted for child porno-
graphy.</h></pair>
```

In other cases, the presence of negation was accidental but we assigned wrongly high costs to edit operations generating *false negatives*. As introduced before, exploiting syntactic information deriving from marking as negated the entire scope of the negation would improve the precision of the system.

The second run was performed by a combined system, that joins the results of EDITS-neg (the system used in our first run) to the results of a more general system that considers all but the negation phenomena (EDITSall-but-neg). The two runs do not differ significantly, even if some improvements can be noticed. We expect that adding more specialized engines, and therefore taking into consideration a higher number of linguistics aspects, will enable us to obtain better results.

For the third run we used the distances calculated by several distance algorithms and cost estimation functions combined using Support Vector Machine learning algorithm to recognize entailment. The distance algorithm used are: Linear Distance, Tree Edit Distance, Longest Common Subsequence and Word Overlap. We use this run to test whether different approaches for the calculation of edit costs can perform in complementary manner. Although for the time being this system performs better than the combined system 2, the accuracy on RTE4 was lower than the accuracy calculated on the previous versions of RTE datasets. This can be caused by the fact that this system considers only lexical similarities, therefore the accuracy on a completely unknown test set (not related to a provided training set, as in the previous RTE Challenges) can decrease.

6 Conclusions and future works

We have presented an approach for RTE based on different independent specialized entailment engines (EDITS), each of which able to deal with a specific phenomena of the language variability.

In the future we plan to extend the model adding other specialized engines, focusing on other linguistic phenomena such as the modals or the active/passive syntactic construction. Future work will also concentrate on the open issue of how to combine the specialized entailment engine according to the dependencies of the linguistic phenomena involved.

Acknowledgements. This work has been developed under the EU-FP6 QALL-ME project.

References

Peter Clark, William R. Murray, John Thompson, Phil Harrison, Jerry Hobbs, Christiane Fellbaum. 2007. *On the Role of Lexical and World Knowledge in RTE3*. Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. 28-29 June 2007, Prague (Czech Republic).

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech and Communication. MIT Press.

Sanda M. Harabagiu, Andrew Hickl, and V. Finley Lacatusu. 2006. *Negation, Contrast and Contradiction in Text Processing*. AAAI.

Milen Kouylekov, and Bernardo Magnini. 2006. *Combining Lexical Resources with Tree Edit Distance for Recognizing Textual Entailment*. In Quiñonero-Candela et al., editors, MLCW 2005, LNAI Volume 3944. Springer-Verlag.

MTed Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. *Word-Net::Similarity - Measuring*

the Relatedness of Concepts.. In In Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NACL-2004). May 2004, Boston, MA.

Emanuele Pianta, Christian Girardi, Roberto Zanoli. 2008. *The TextPro tool suite*. Proceedings of LREC, 6th edition of the Language Resources and Evaluation Conference. 28-30 May 2008, Marrakech (Morocco).

Rion Snow, Lucy Vanderwende, and Arul Menezes. 2006. *Effectively using syntax for recognizing false entailment*. Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. New York.

Marta Tatu, Dan I. Moldovan. 2007. *COGEX at RTE 3*. Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. 28-29 June 2007, Prague (Czech Republic).

Lucy Vanderwende, Arul Menezes, Rion Snow. 2006. *Microsoft Research at RTE-2: Syntactic Contributions in the Entailment Task: an implementation*. Proceedings of the Second PASCAL Recognizing Textual Entailment Challenge. 10 April 2006, Venice (Italy).

Kaizhong Zhang, and Dennis Shasha. 1990. *Fast Algorithm for the Unit Cost Editing Distance Between Trees*. Journal of algorithms, vol 11, December, 1990.