UAIC Participation at RTE-6

Adrian Iftene, Mihai-Alex Moruz

"Al. I. Cuza" University, Faculty of Computer Science, Iasi, Romania

{adiftene, mmoruz}@info.uaic.ro

Abstract

Textual entailment recognition is the task of deciding, given two text fragments, whether the meaning of one text can be deduced from the other. This year, at our fourth participation in the RTE competition, we improved the system built for the RTE-5 competition. Our system solves entailment by attempting to map every word in the hypothesis to one or more words in the text. For that, we transform the hypothesis, using extensive semantic knowledge from sources like DIRT, VerbNet, WordNet, VerbOcean, Wikipedia and the Acronym database.

1. Introduction

The Recognizing Textual Entailment (RTE) task consists of creating a system that, given two pieces of text, can determine if the meaning of one text is entailed, or can be deduced from the other text. Since the first edition in 2005, RTE has steadily grown importance in the NLP community, as it appears to work as a common framework in which to analyze, compare and evaluate different techniques used in NLP applications to deal with semantic inference, a common issue shared by many NLP applications. The first three PASCAL RTE Challenges campaigns were held in Europe, and, starting with 2008, RTE became a track at the Text Analysis Conference (TAC 2008), and thus came into contact with other communities working on significantly different NLP applications such as knowledge extraction and summarization. The interaction has provided the opportunity for the application of RTE systems to various settings and has pointed the RTE community towards more realistic scenarios. In particular, the RTE-5 Pilot Search Task represented a step forward, as for the first time textual entailment recognition was performed on a real text corpus. Furthermore, it was set up in the Summarization setting, attempting to analyze the potential impact of textual entailment on a real NLP application.

According to the RTE-6[1] guidelines, by capitalizing on the promising outcome of the RTE-5[2] Pilot Search Task, RTE-6 has two goals:

- *to advance the state of the art in RTE*, by proposing a data set which reflects the natural distribution of entailment in a corpus and presents all the problems that can arise while detecting textual entailment in a natural setting, such as the interpretation of sentences in their discourse context;
- *to further explore the contribution that RTE engines can make to Summarization applications*. In a general summarization setting, correctly extracting all the sentences entailing a given candidate statement for the summary (similar to Hypotheses in RTE) corresponds to identifying all its mentions in the text, which is useful to assess the importance of that candidate statement for the summary and, at the same time, to detect those sentences which contain redundant information and should probably not be included in the summary. Furthermore, if automatic summarization is performed in the Update scenario (where systems are required to write a short summary of a set of newswire articles, under the assumption that the user has already read a given set of earlier articles) it is important to distinguish between novel and non-novel information. In such a setting, RTE engines which are able to detect the novelty of H's can help Summarization systems filter out non-novel sentences from their summaries.

The basis for the system used in RTE-6 was the system created for RTE-5 (Iftene and Moruz, 2009), which was further refined and honed. Since last year's system was the winning entry, no significant changes have been made to its architecture; most of the modifications consist of tweaking various thresholds and extra pre-processing of the pairs. The system consists of a preprocessing module and a decision module, which contains the rules that determine the value of the entailment pair. A schematic of the system is given in Figure 1 below. The basis of the RTE-5 system is described in sections 2 and 3, along with the improvements made for RTE-6. Broadly, the system first runs the input through a preprocessing stage, which expands contractions, recognizes NEs, performs dependency parsing and recognizes dates. The processed input is then fed into the decision module, which, using resources such as WordNet, VerbOcean, VerbNet, and so on, makes a decision upon the entailment value of the given pair.

[1] RTE-6: http://www.nist.gov/tac/2010/RTE/

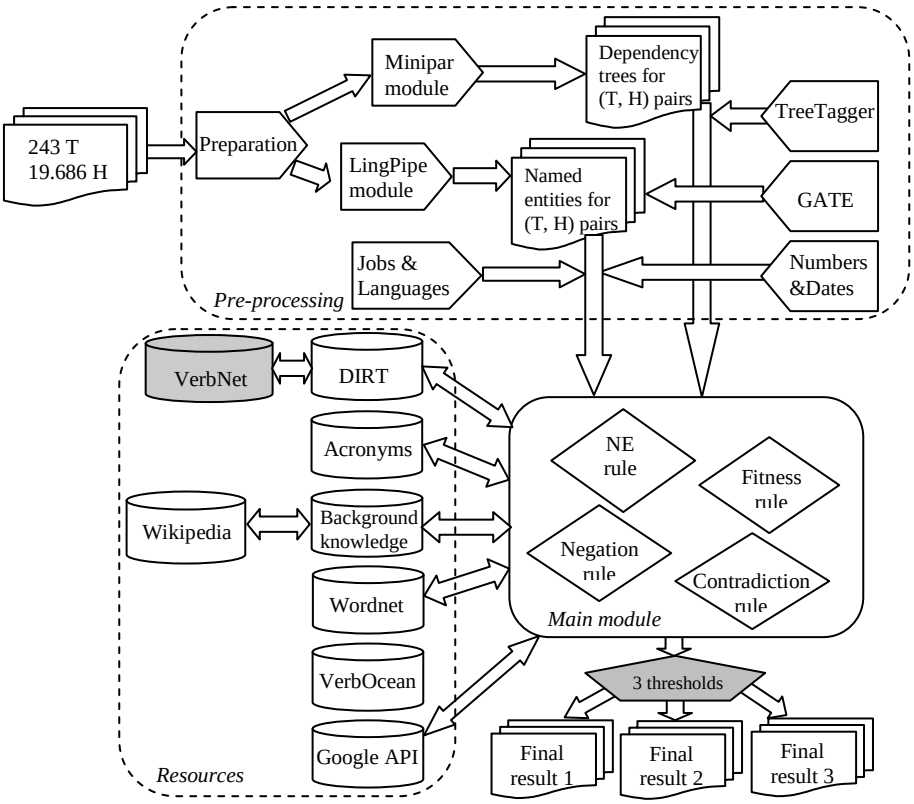

Figure 1: RTE-6 System architecture

2. Pre-Processing

In order to improve the results of the preprocessing step, some modifications are performed (Iftene, 2009). Thus, in all data we replace "*hasn't*" with "*has not*", "*isn't*" with "*is not*", "*couldn't*" with "*could not*", etc. The meaning remains the same after this transformation, but the quality of the MINIPAR (Lin, 1998) output is considerably improved. Also, before sending the text to LingPipe[3], we replace some punctuation signs like quotation marks "", brackets (), [], {}, commas, etc. with the initial sign padded with space characters. Again, the meaning of the text is the same, but the quality of the LingPipe output is better after this transformation.

After the preparation step, the text and the hypothesis are parsed with MINIPAR (Lin, 1998). For those cases in which MINIPAR does not identify any verb in the processed sentence, we use TreeTagger for identifying, with a higher degree of precision, the Part-Of-Speech (POS) and replace the incorrect POS determined by MINIPAR. This step is very important, especially in the case

[2] RTE-5: http://www.nist.gov/tac/2009/RTE/
[3] LingPipe: http://www.alias-i.com/lingpipe/

of verbs, because our algorithm begins its mapping with verbs, and builds the comparison on the basis of verb arguments.

In parallel, the result obtained after preparation is processed by LingPipe, in order to identify named entities (NEs). In the case of the Named Entities of type JOB and LANGUAGE, we additionally used GATE (Cunningham et al., 2001), whose gazetteer contains finer-grained classes of entities, which considerably increase the accuracy of NE extraction.

3. Main Module

The purpose of the main module is to map all the nodes from the hypothesis syntactic tree to at least one node in the syntactic tree of the text, in a similar manner as that described in (Iftene, 2008). The mapping between entities can be done either *directly* (when entities from hypothesis tree exist in the text tree) or *indirectly* (when entities cannot be mapped directly and require transformations using external resources). The mapping of each node yields a fitness value, on the basis of the transformations involved in order to carry it out (exact match yields a fitness of 1, for example, and antonyms yield a fitness of -1), which indicates the similarity between entities of the text and the hypothesis. Using the local fitness values, we build an extended local fitness and then, using all partial values, we calculate a normalized value that represents the global fitness. When an entity from the hypothesis can be mapped to more entities from the text, we select the mapping which maximizes global fitness.

The global fitness value is then used to determine the relation between text–hypothesis pairs. The "*No entailment*" cases are represented by pairs for which the global fitness value is below a threshold, the value of which is extracted from the training data, and the "*Entailment*" cases are represented by pairs for which global fitness is above the same threshold; for separating contradiction and unknown cases, we considered another threshold, also extracted from the training data.

3.1. Entailment Cases

Basic Positive Rules

In order to determine the global fitness for a given entailment pair, we attempt a mapping of the nodes from the syntactic tree of the hypothesis to the nodes of the syntactic tree of the text. For every node from the hypothesis tree which can be mapped directly to a node from the text tree, we

consider the local fitness value to be 1 (which represents the maximum value). When direct mapping is not possible, we apply transformations, based on external knowledge sources, to the hypothesis node so that it becomes more similar to some node in the text. For verbs we use DIRT (Lin and Pantel, 2001) and transform the hypothesis tree into an equivalent one, where the verb node is replaced with an equivalent form. This is supplemented by the use of VerbNet[4] in order to determine the relation between the verbs in the hypothesis and the text, as described in (Moruz, 2010). This is the case of the example below, where in the text we have "*An English-born blues legend, passed away…*" and in the hypothesis we have "*A musician has died…*". After using this resource, the hypothesis changes into "*A musician passed away*" and in this form it is easier to compare the text and hypothesis and, in the end, the value of the global fitness score is increased.

In the case of named entities, we either use an acronym database or obtain information related to it from background knowledge (Iftene and Balahur, 2008). Apart from the relations between acronyms and full names, we also take into account relations such as part-of (relation between Basel in Switzerland and European city), etc.

For nouns and adjectives we use WordNet (Fellbaum, 1998) and some of the relations from eXtended WordNet[5] to look up synonyms, which we then attempt to map to nodes from the text tree.

For every transformation with DIRT or WordNet, we will consider the similarity value indicated by these resources the value of local fitness. When we use the acronym database or background knowledge we consider the local fitness to be 1.

Positive Rules for Numbers

In the case of numerical data the mapping process is not as straightforward as for nouns, for example, and some special situations need to be taken into account. There are cases in which, even if the numbers from the text and the hypothesis do not correspond, certain quantifiers may change their meaning enough for a positive match. For solving these cases, we create intervals for both the text and the hypothesis expressions; if the interval from the text is contained in the interval from the hypothesis, we award a local fitness value of 1. The quantifiers are taken from a list which contains expressions such as "*more than*", "*less than*", or words such as "*over*", "*under*", etc.

[4] VerbNet: http://verbs.colorado.edu/~mpalmer/projects/verbnet.html
[5] eXtended WordNet: http://xwn.hlt.utdallas.edu/

3.2. No Entailment Cases

Basic Negative Rules

If after all checks are made we cannot map a node from the hypothesis syntactic tree, we insert a penalty into the value of the node's local fitness. Also, because the stop words from the hypothesis ("*the*", "*an*", "*a*", "*at*", "*to*", "*of*", "*in*", "*on*", "*by*", etc.) artificially increase the value of global fitness, we do not take them into consideration in the final global fitness calculation.

Negation Rules

For every verb from the hypothesis we consider a Boolean value which indicates whether the verb is negated or not. For determining negation, we examine the verb's subtree for words such as "*not*", "*never*", "*may*", "*might*", "*cannot*", etc. For each of these words we successively negate the initial truth value of the verb, which by default is "*false*".

A specific rule was also built for verbs in the infinitive (usually preceded by the particle "*to*"). In this case, the sense of the verb is strongly influenced by the active verb, adverb or noun preceding the particle "*to*", as follows: if it is being preceded by a verb like "*believe*", "*glad*", "*claim*" or their synonyms, an adjective like "*necessary*", "*compulsory*", "*free*" or their synonyms or a noun like "*attempt*", "*trial*" and their synonyms, the meaning of the verb in infinitive form is stressed upon and becomes "*certain*". For all other cases, the "*to*" particle diminishes the certainty of the action expressed in the infinitive-form verb.

Contradiction Cases

For determining contradiction, we consider several situations, the most common of which is the negation of the verb with words like "*never*", "*not*", "*no*", "*cannot*", "*unsuccessfully*", "*false*" etc. In the example below, the text is "*Movie studio company, New Line Cinema has announced that movie director Peter Jackson will never be allowed to work on another New Line films*." and in the hypothesis "*New Line wants to work with Peter Jackson*.".

Another type of contradiction case is that of long infinitive verbs preceded by words such as "*refuse*", "*deny*", "*ignore*", "*plan*", "*intend*", "*proposal*", "*able*", etc.

Contradiction is also determined on the basis of the *antonymy* relation between words from the text and the hypothesis. For determining antonymy, we use the [*opposite-of*] relation from VerbOcean (Chklovski and Pantel, 2004) and antonymy relation from WordNet. The domain of the antonymy relation is broadened by combining synsets and antonyms in WordNet or opposites from VerbOcean. For words from the hypothesis which cannot be mapped to words from the text using either synonymy or antonymy, we consider the set of antonyms for their synonyms and then check if any word from this new set cans me mapped to the text.

In some situations, the similarity relation from DIRT is an antonymy relation (the scores in DIRT are in fact more similar to co-occurrence scores), and for this reason we do an extra verification of DIRT relations to see if we have antonymy in either WordNet or VerbOcean. For all identified contradiction cases, since we consider the penalties with the highest values, the final answer for the considered pairs will be "*Contradiction*".

Unknown Cases

If the text or hypothesis contains words such as "*may*", "*can*", "*should*", "*could*", "*must*", "*might*", "*infrequent*", "*rather*", "*probably*", etc., the penalties are not decisive in establishing the final answer, which is obtained only after computing global fitness. If the score is not low enough, the solution for the entailment pair is "*Unknown*".

With regards to the particle "*to*" we will consider as "*Unknown*" those cases which are not determined to be contradictions.

In the case of named entities, however, the solution we have chosen is different. If even after using the acronym database we cannot map an entity from the hypothesis to an entity in the text, we decide that a pertinent conclusion cannot be drawn, and the result for the pair is "*Unknown*".

If any of the numbers in the text or the hypothesis has an attached unit of measure, it is always kept, as it is possible to find the same numbers in the text and the hypothesis, but to have those numbers referring to different entities:

```
T: At least 14 people have been killed in a suicide bomb attack; government officials were among the 35 injured.

H: 35 government officials were injured.
```

An exception to the named entity rule presented above is the case when the entity is a first name, in which case we only insert a penalty in the global fitness:

```
T: A man is accused of killing Ms. Zapata.

H: Angie Zapata has been killed.
```

For the Main Task in the RTE-6 challenge it was no longer necessary to separate the no entailment cases into "*Unknown*" and "*Contradiction*". Because of this, the threshold separating these two sets of pairs is ignored, and the same answer is given in both cases, but it is important to note that the option to separate unknown and contradiction is still available.

4. Results in RTE-6

Using the system described in section 3, we participated in the main and novelty detection tasks of the RTE-6 evaluation campaign with three distinct runs, obtained by running the system with different thresholds. The results are given in table 1 below:

Run ID	Precision	Recall	F-Measure
001	22.89%	27.20%	24.85%
002	14.02%	39.15%	20.64%
003	31.49%	17.46%	22.46%

Table 1: Results for the RTE-6 Main Task

The first run was obtained with the thresholds set to maximize both precision and recall. Run two was obtained by lowering the threshold for separating the entailment and non-entailment cases; this is the reason for the higher recall and the lower precision. The third run was obtained by raising the value of the threshold and thus the justification for the "mirrored results".

The average, top and bottom results in the Main task are 33.72%, 48.01% and 11.60% respectively. The lower results we obtained this year are due to the significant change in datasets; also, we did not take into account the information available in discourse, due to the lack of tools to perform such analyses (we did not include a coreference engine, for example).

The results for the novelty task are given in table 2 below:

Run ID	Precision	Recall	F-Measure
001	81.40%	70.00%	75.27%
002	81.54%	53.00%	64.24%
003	73.28%	85.00%	78.70%

Table 2: Results for the RTE-6 Novelty Detection Task

The runs for this task were obtained by running the systems with the settings described above; the reason for run 3 having the highest score is the difference in the scoring method. The average, top and bottom scores for this task were 77.84%, 82.91% and 43.98% respectively. The reason for which the third configuration of the system performed better for the novelty detection subtask is that most of the sentences were not novel. Because of this, by raising the entailment threshold, we reduced the number of false positives, and thus raised our recall greatly, while only slightly lowering our precision.

5. Ablation Tests

In order to determine each component's relevance, the system was run in turn with each component removed (Iftene, 2009). This technique was first employed for the RTE-3 system and was used after that in RTE-4 and in RTE-5. Table 3 presents these results for the system used in RTE-6, where the meanings for P, C and WR are: P = *Precision*, R = *Recall*, F = *F-measure*, C = *Contribution* and WR = *Weighted Relevance* (Contribution and Weighted relevance are computed with regard to f-measure). We will only present the ablation test results for our best run, the first.

System Description	RTE-6				
	P (%)	R (%)	F (%)	C (%)	WR (%)
Without DIRT	25.86	26.98	26.41	-1.56	-6.27
Without BK	23.91	22.01	22.92	1.93	7.76
Without the NE rule	25.86	26.98	26.41	-1.56	-6.27
Without the Negation rule	22.67	28.25	25.15	-0.30	-1.2
Without the Contradiction rule	22.87	27.30	24.89	-0.04	-0.16

Table 3: Components' relevance for 2-way task

The meanings of the columns are the following:

- $Precision_{Without\_Component}$ value was obtained by running the system without a specific component (for example, $Precision_{Without\_DIRT}$ is 25.86% and it represents the precision of the system without the DIRT component);
- $Recall_{Without\_Component}$ value was obtained by running the system without a specific component (for example, $Recall_{Without\_DIRT}$ is 26.98% and it represents the precision of the system without the DIRT component);
- $F\text{-}measure_{Without\_Component}$ value was obtained by running the system without a specific component (for example, $F\text{-}measure_{Without\_DIRT}$ is 26.41% and it represents the precision of the system without the DIRT component);
- $Contribution_{Component} = Full\_system\_F\text{-}measure - F\text{-}measure_{Without\_Component}$ (for example, $Contribution_{DIRT}$ is 24.85 % - 26.41% = -1.56 % for the DIRT component of the system, where 24.85 is the f-measure for the full system and 26.41% is the f-measure for RTE-3 system without DIRT component);
- $WeightedRelevance_{Component} = \frac{100 \times Contribution_{Component}}{Full\_system\_f-measure}$ (for example, for the DIRT component, $WeightedRelevance_{DIRT} = \frac{100 \times Contribution_{DIRT}}{Full\_system\_precision} = = \frac{100 \times -1.56}{69.13} = -6.27\%$).

As can be seen in Table 3, most of the components actually decrease the performance of the system for Run 1. We have not yet been able to determine the reason for this, and we intend to carry out further analysis over the data in order to understand this.

From the ablation tests for Run 2, we see how the following components increase the performance of the system: Background Knowledge (with 1.08 %), NE rule with (with 4.45%) and contradiction rule (with 0.56%).

Similarly, from the ablation tests for Run 3, we see how the following components increase the performance of the system: Background Knowledge (with 1.30 %) and contradiction rule (with 0.01%).

6. *Conclusions*

The results for this year's competition are not as good as those from last year, mostly because of the significant difference in datasets. Further analysis of the results is necessary, in order to improve the actual results. We have also prepared ablation tests, which are runs of the system with resources left out, in order to determine the increase in performance each component brings, but the results are still to be released at this point.

Because of limitations in availability of resources for discourse parsing (coreference engine, for example) and because of the lack of a framework for analyzing large numbers of large texts in a timely manner, we were unable to participate in the pilot task defined for this year's competition. However, analysis of the data provided by the organizers has led us to a series of important intuitions and ideas for applying textual entailment for various real world applications, such as opinion mining.

As future work, we intend to determine the main cause for the decrease in performance; also, we need to extend the use of verb knowledge in the process of determining textual entailment in order to increase the performance of the system.

The main goal now is to attempt to apply textual entailment to solving real world problems, such as validating information extraction and opinion mining.

Acknowledgments

The research presented in this paper was funded by the Sector Operational Program for Human Resources Development through the project "Development of the innovation capacity and increasing of the research impact through post-doctoral programs" POSDRU/89/1.5/S/49944. The authors thank the members of the NLP group in Iasi for their help and support at different stages of the system development.

References

Chklovski, T., Pantel, P. 2004. *VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations*. In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04). Barcelona, Spain.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V. 2001. *GATE: an architecture for development of robust HLT applications.* In ACL '02: Proceedings of the 40th Annual Meeting on

Association for Computational Linguistics, 2001, 168--175, Association for Computational Linguistics, Morristown, NJ, USA.

Fellbaum, C. 1998. *Wordnet: An electronic lexical database*. MIT Press, Cambridge, Mass.

Iftene, A. 2008. *UAIC Participation at RTE4*. In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, 2008. Gaithersburg, Maryland, USA.

Iftene, A. 2009. *Textual Entailment*. PhD Thesis. "Al. I. Cuza" University. March 13, 2009. Iasi, Romania. (http://thor.info.uaic.ro/~adiftene/thesisAI.pdf)

Iftene, A., Balahur-Dobrescu, A. 2008. *Named Entity Relation Mining Using Wikipedia*. In Proceedings of the Sixth International Language Resources and Evaluation (LREC'08). 28-30 May, Marrakech, Morocco.

Iftene, A., Trandabăţ, D., Pistol, I., Moruz, A., Husarciuc, M., Cristea, D. 2009. *UAIC Participation at QA@CLEF2008*, Evaluating Systems for Multilingual and Multimodal Information Access, Lecture Notes in Computer Science, vol. 5706/2009, pp. 385-392, ISBN 978-3-540-74998-1, ISSN 0302-9743 (Print) 1611-3349.

Iftene, A., Moruz M. A. 2009. *UAIC Participation at RTE-6*, In Text Analysis Conference (TAC 2009) Workshop - RTE-5 Track. National Institute of Standards and Technology (NIST). November 16-17, 2009. Gaithersburg, Maryland, USA.

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

Lin, D., Pantel, P. 2001. *DIRT - Discovery of Inference Rules from Text*. In Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD-01). pp. 323-328. San Francisco, CA.