# UAIC Participation at AVE 2007

Adrian Iftene[1] and Alexandra Balahur-Dobrescu[1,2]

[1] UAIC: Faculty of Computer Science, “Alexandru Ioan Cuza” University, Romania
[2] University of Alicante, Department of Software and Computing Systems,
Alicante, Spain
{adiftene,abalahur}@info.uaic.ro

**Abstract.** Textual Entailment Recognition (RTE) is a recently proposed task, aiming at capturing the means through which textual inferences can be made. Moreover, using such a module is meant to contribute to the increase in performance of many NLP applications, such as Summarization, Information Retrieval or Question Answering, both for answer ranking as well as for answer validation. This article presents the manner in which we used the TE system built for the RTE3 competition this year for the AVE exercise. We describe the steps followed in building the patterns for question transformation, the generation of the corresponding hypotheses and finally for answer ranking. We conclude by presenting an overview of the performance obtained by this approach and a critical analysis of the errors obtained.

## 1 Introduction

AVE[1] is a new task introduced in the QA@CLEF competition, with the aim of promoting the development and evaluation of subsystems validating the correctness of the answers given by QA systems. In this task, the systems must emulate human assessment of QA responses and decide whether an Answer to a Question is correct or not according to a given Text.

Participant systems receive a set of triplets (Question, Answer, and Supporting Text) and they must return a Boolean value for each triplet. Results are evaluated against by the QA human assessments.

The AVE task described is hence similar to the Recognizing Textual Entailment task (Dagan et al., 2006), for which systems are built and a Recognizing Textual Entailment competition has been organized for the last three years by the PASCAL excellence network for multimodal interfaces.

Formally, textual entailment is defined (Dagan et al., 2006) as a directional relation between two text fragments, termed *T* - the entailing text, and *H* - the entailed text. It is then said that *T* entails *H* if, typically, a human reading *T* would infer that *H* is most likely true. Textual entailment recognition is thus the task of deciding, given *T* and *H*, whether *T* entails *H*. In the textual entailment competition, participants are provided

[1] http://nlp.uned.es/QA/ave/

C. Peters et al. (Eds.): CLEF 2007, LNCS 5152, pp. 395–403, 2008.

with pairs of small text snippets and they must build a system that should judge the truth value of the entailment relation for each pair.

## 2 Textual Entailment System

The main idea of our system is to transform the hypothesis making use of extensive semantic knowledge from sources like DIRT, WordNet, Wikipedia, and database of acronyms. Additionally, we built a system to acquire the extra Background Knowledge needed and applied complex grammar rules for rephrasing in English.

### 2.1 Tools

**LingPipe**
The first step splits the initial file into pairs of files of text and hypothesis. All these files are then sent to the LingPipe[2] module in order to find the Named entities.

**MINIPAR**
In parallel, we transform with MINIPAR[3] (Lin, 1998) both the text and the hypothesis into dependency trees. For every node from the MINIPAR output (which represents a simple word belonging to a sentence), we consider a stamp called **entity** with three main features: the node lemma, the father lemma and the edge label.

### 2.2 Resources

The main module receives separate files for each text and hypothesis pair from the initial data (test or development). The remaining resources used are DIRT, WordNet, the Acronyms Database and the Background Knowledge. They are used to expand each remaining word from the hypothesis to a list of similar or related terms and thus increase the probability to find the initial word or any of its equivalents in the list of words from the text.

**The DIRT resource:** DIRT[4] (Discovery of Inference Rules from Text) is both an algorithm and a resulting knowledge collection created by Lin and Pantel at the University of Alberta (Lin and Pantel, 2001), (Lin, 2001). A path, extracted from a parse tree, is an expression that represents a binary relationship between two nouns. For the hypothesis verbs in the MINIPAR output without correspondence, we extract templates with DIRT like format. In the same way, we build a list with templates for the verbs in the text tree. With these two lists of templates we perform a search in the DIRT database and extract the “best” trimming using template type (full or partial) and the DIRT score.

[2] http://www.alias-i.com/lingpipe/

[3] http://www.cs.ualberta.ca/~lindek/minipar.htm

[4] http://aclweb.org/aclwiki/index.php?title=DIRT_Paraphrase_Collection

**eXtended WordNet:** eXtended WordNet[5] is an ongoing project at the Human Language Technology Research Institute, University of Texas at Dallas. In the eXtended WordNet, the WordNet glosses are syntactically parsed transformed into logic forms and content words are semantically disambiguated. For non-verbs nodes from the hypothesis tree, if in the text tree we do not have nodes with the same lemma, we search for their synonyms in the extended WordNet.

The **acronyms' database**[6] helps our program to find relations between the acronym and its meaning: "US - United States".

**The Background Knowledge** was used in order to expand the named entities and the numbers. It was built semi-automatically, and it used a module in which language could be set according to the current system working language and thus the corresponding Wikipedia[7] could be selected. For every named entity or number in the hypothesis, the module extracted from Wikipedia a file with snippets with information related to them.

For every node transformed with DIRT or with the eXtended WordNet, we consider its **local fitness** as being the similarity value indicated by DIRT or by eXtended WordNet. In other cases, when there is a direct mapping or when we use the acronyms database or the Background Knowledge, we consider the **local fitness** of the node to be 1.

## 2.3 Rules

**Semantic Variability Rules:** negations and context terms
For every verb from the text and hypothesis we consider a Boolean value which indicates whether the verb has a negation or not, or, equivalently, if it is related to a verb or adverb **diminishing** its sense or not. For that, we use the POS-tags and a list of words we consider as introducing a negation: "*no*", "*don't*", "*not*", "*never*", "*may*", "*might*", "*cannot*", "*should*", "*could*", etc. For each of these words we successively negate the initial truth-value of the verb, which by default is "*false*". The final value depends on the number of such words.

**Rule for Named Entities** from hypothesis without correspondence in text Additionally, we have a separate rule for named entities from the hypothesis without correspondence in the text. If the word is marked as named entity by LingPipe, we try to use the acronyms' database or obtain information related to it from the background knowledge. In the event that even after these operations we cannot map the word from the hypothesis to one word from the text, we increase a value that counts the problems regarding the named entities in the current pair. We then proceed to calculating a fitness score measuring the syntactic similarity between the hypothesis and the text, further used as one of the features that the two classifiers used are trained on.

[5] http://xwn.hlt.utdallas.edu/

[6] http://www.acronym-guide.com

[7] http://en.wikipedia.org/wiki/Main_Page

**Rule for determination of entailment**

After transforming the hypothesis tree, we calculate a global fitness score using the following **extended local fitness** value for every node from the hypothesis - which is calculated as sum of the following values:

1. local fitness obtained after the tree transformation and node mapping,
2. parent fitness after parent mapping,
3. mapping of the node edge label from the hypothesis tree onto the text tree,
4. node position (left, right) towards its father in the hypothesis and position of the mapping nodes from the text.

We calculate for every node from the hypothesis tree the value of the extended local fitness, and afterwards consider the normalized value relative to the number of nodes from the hypothesis tree. We denote this result by *TF* (*Total Fitness*). After calculating this value, we compute a value *NV* (the negation value) indicating the number of verbs with the same value of negation. Because the maximum value for the extended fitness is *4*, the complementary value of the *TF* is *4-TF*. The formula for the **global fitness** used is therefore:

$$GlobalFitness = NV * TF + (1 - NV) * (4 - TF)$$

Using the development data, we establish a threshold value of 2.06, and according to this, we decide that pairs above it will have the answer "yes" for entailment.

## 2.4 Results in RTE3

We submitted two runs for our system, with the difference residing in the parameters used in calculating the extended local fitness, but with the same final score: 69.13 %.

To be able to see each component's relevance, the system was run in turn with each component removed. The results in the table below show that the system part verifying the NEs is the most important.

**Table 1.** Components relevance

| System Description | Precision | Relevance |
|---|---|---|
| Without DIRT | 0.6876 | 0.54 % |
| Without WordNet | 0.6800 | 1.63 % |
| Without Acronyms | 0.6838 | 1.08 % |
| Without BK | 0.6775 | 2.00 % |
| Without Negations | 0.6763 | 2.17 % |
| Without NEs | 0.5758 | 16.71 % |

Twenty-six teams participated in the third challenge, and even though this was our first participation in the RTE competition, our system was ranked third, being among the best in the competition.

## 3 Using the TE System in the AVE track

The data provided in the AVE task are in the following format:

**Table 2.** Question 148 from the test data

```
<q id="148" lang="EN">
        <q_str>When was Yitzhak Rabin born?</q_str>
        <a id="148_1" value="">
                <a_str>March 1 1922</a_str>
                <t_str doc="GH951109-000097"> Yitzhak Rabin, Prime Minister of Israel;
born Jerusalem, March 1, 1922, died Tel Aviv, November 4, 1995. …
       </t_str></a>
        <a id="148_2" value="">
                <a_str>1992-1995</a_str>
                <t_str doc="en/p03/368881.xml">Yitzhak Rabin 1992-1995</t_str></a>
        <a id="148_4" value="">
                <a_str>4 19</a_str>
                <t_str doc="168208.xml">Yitzhak Shamir   Yitzhak Shamir 7   יִצְחָק שָׁמִיר
Prime Minister of Israel  …. </t_str></a>
        <a id="148_5" value="">
                <a_str>1995</a_str>
     <t_str doc="650871.xml">Ministry of Interior (Israel) ……</t_str> </a>
   </q>
```

The system architecture is presented below:

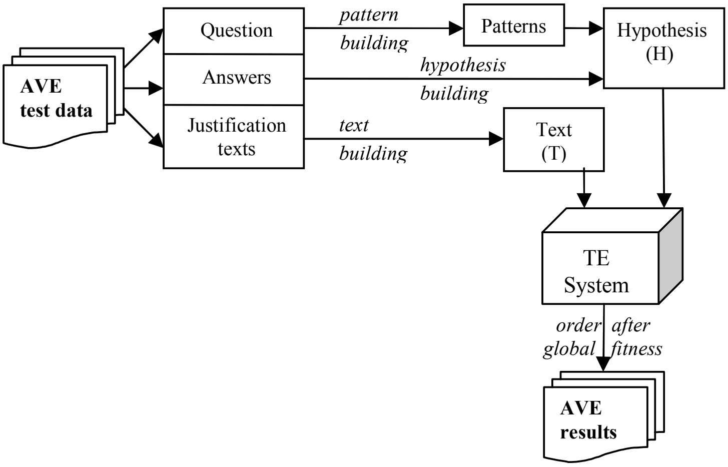

**Fig. 1.** AVE System

The steps executed by our system are as follows:

- We build a pattern with variables for every question according to the question type;
- Using a pattern and all possible answers, we build a set of hypotheses for each of the questions: $H_1$, $H_2$, $H_3$ etc.;
- We assign the justification snippet the role of text T and we run the TE system for all obtained pairs: ($T_1$, $H_1$), ($T_2$, $H_2$), ($T_3$, $H_3$), etc.

Lastly, we consider the correct answer for the current question the candidate from the hypothesis for which we obtain the greatest global fitness.

### 3.1 Pattern Building

In order to use the TE system for ranking the possible answers in the AVE task, all these questions are first transformed according to the algorithm presented in (Bar-Haim et al., 2006).

For question 148 we have:

**Question:** *When was Yitzhak Rabin born?*

Our program generates the following pattern:

**Pattern:** *Yitzhak Rabin was born at DATE.*

where *DATE* is the variable in this case. We generate specific patterns according to the following answer types: Measure, Person, Other, Location, Organization and Date. Next table presents the identified types of patterns:

**Table 3.** Examples of Patterns

| Answer type | Cases Number | Question example | Pattern |
|---|---|---|---|
| Date | 3 | When was Yitzhak Rabin born? | Yitzhak Rabin was born **at DATE**. |
| Measure | 5 | How many bush fires were there near Sydney in January 1994? | **MEASURE** bush fires were there near Sydney in January 1994. |
| Location | 6 | Where is the Hermitage Museum? | The Hermitage Museum is **in LOCATION**. |
| Person | 12 | Who wrote the song "Dancing Queen"? | **PERSON** wrote the song “Dancing Queen”. |
| Organization | 17 | What company makes Ribena? | **ORGANIZATION** company makes Ribena. |
| Other | 23 | What is Gulf War Syndrome? | Gulf War Syndrome is **OTHER**. |

Following the building of the pattern, we proceed to constructing the corresponding hypotheses.

### 3.2 Hypothesis Building

Using the pattern building mechanism above and the answers provided within the AVE data, we built the corresponding hypotheses. For example, for question 148, we built, according to the answers from the English test data (“a_str” tags), the following hypotheses:

$H_{148\_1}$: *Yitzhak Rabin was born at March 1 1922.*

$H_{148\_2}$: *Yitzhak Rabin was born at 1992-1995.*

$H_{148\_3}$: *Yitzhak Rabin was born at 4 19.*

$H_{148\_4}$: *Yitzhak Rabin was born at 1995.*

For each of these hypotheses, we consider as having the role of text *T* the corresponding justification text (content of “t_str” tag).

### 3.3 Answers Classification

We consider the pairs built above as input for our Textual Entailment system. After running the TE system, the global fitness values for these pairs are the following:

$GlobalFitness(H_{148\_1}, T_{148\_1}) = 2.1148$

$GlobalFitness(H_{148\_2}, T_{148\_2}) = 1.8846$

$GlobalFitness(H_{148\_3}, T_{148\_3}) = 2.1042$

$GlobalFitness(H_{148\_4}, T_{148\_4}) = 1.7045$

Since in the considered case the highest value is obtained for the answer *March 1 1922,* we consider it as the SELECTED answer and the rest as VALIDATED. The REJECTED answers were considered the pairs for which we have NE problems (in this case, the global fitness has the minimum value, i.e. 0).

### 3.4 Results and Errors Analysis

Our AVE system has the following results:

**Table 4.** AVE Results

| F measure | 0.34 |
|---|---|
| Precision over YES pairs | 0.21 |
| Recall over YES pairs | 0.81 |
| QA accuracy | 0.21 |

Our results as compared to other participants are presented below (Peñas et al., 2007):

We compare our results against the gold file and count to see which class of answers our correct and incorrect answers pertain to. In table below we can observe the fact that most problems arose within the REJECTED class. The cause of this issue was that our TE system considers as being rejected only the pairs for which NE problems were encountered (in this case, the global fitness is zero). This rule functions very well for 111 cases from 174, but as it can be observed, it is not enough. In all

**Table 5.** Comparing AV systems performance with QA systems in English for first 5 systems

| Group | QA accuracy | % of perfect selection |
|---|---|---|
| DFKI 2 | 0.21 | 70% |
| **UAIC Iasi** | **0.21** | **70%** |
| UA 2 | 0.19 | 65% |
| U.Indonesia | 0.18 | 60% |
| UA 1 | 0.18 | 60% |

other cases, we calculate the global fitness, and the answer with the highest score is considered SELECTED and all other answers are considered as VALIDATED. One solution for this problem was to train our TE system on the AVE development data and identify a specific threshold according to the AVE input data.

**Table 6.** Distribution on answers classes of our Results

| Answers Class in Gold file | Unknown | Valid`ated | Rejected | Total |
|---|---|---|---|---|
| Correct | | 17 | 111 | 128 |
| Incorrect | 7 | 4 | 63 | 74 |

## 4 Conclusions

We showed how the TE system used in the RTE3 competition can successfully be used as part of the AVE system, resulting in improved ranking between the possible answers, especially in the case of questions with answers of type Person, Location, Date and Organization.

The main problem of our system arises from the rule that identifies the REJECT cases in the AVE competition. We notice that the rule regarding the presence of NEs is very good in this case and identifies 64 % of the correct cases, but it is not enough to identify the entire class of REJECTED answers. In order to better identify these situations, we must additional rules must be added in order to bring the system improvement.

## References

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The Second PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment, Venice, Italy (2006)

Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Quiñonero-Candela, J., et al. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 177–190. Springer, Heidelberg (2006)

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

Lin, D.: LaTaT: Language and Text Analysis Tools. In: Proc. Human Language Technology Conference, San Diego, California (March 2001)

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

Peñas, A., Rodrigo, Á., Verdejo, F.: Overview of the Answer Validation Exercise 2007. In: Working Notes of CLEF 2007, Budapest, Hungary, 19-21 September (2007)