AUEB at TAC 2009

Prodromos Malakasiotis
Department of Informatics
Athens University of Economics and Business
Patission 76, GR-104 34 Athens, Greece

Abstract

This paper describes AUEB's participation in TAC 2009. Specifically, we participated in the textual entailment recognition track for which we used string similarity measures applied to shallow abstractions of the input sentences, and a Maximum Entropy classifier to learn how to combine the resulting features. We also exploited WordNet to detect synonyms and a dependency parser to measure similarity in the grammatical structure of T and H.

1 Introduction

Over the past years, several challenges and workshops concerning subareas of Natural Language Processing (e.g. question answering, textual entailment recognition, summarization etc.) have been organized. This year the National Institute of Standards and Technology (NIST) organized the Text Analysis Conference (TAC) 2009, which is a series of workshops providing the infrastructure for large-scale evaluation of NLP technology. The conference consists of three main tracks, namely question answering, summarization, and textual entailment recognition. In this paper we present AUEB's[1] participation in the textual entailment recognition tracks. In section 2 we provide a brief description of the this year's RTE track. Section 3 describes in detail our system while section 4 presents the official results. Finally, section 5 concludes.

2 Recognizing textual entailment

Textual Entailment is of significant importance in many natural language processing areas, such as question answering, information extraction, information retrieval, and multi-document summarization. In the TAC Recognizing Textual Entailment Challenge (RTE), it is defined as the task of deciding whether or not the meaning of a *hypothesis* text (H) can be inferred from the meaning of another text (T).[2] For instance, the following is a correct entailment pair:

T: Nigeria's Kano State and US drugs firm Pfizer have agreed to settle a multi-million dollar lawsuit out of court, lawyers for both sides say. Pfizer has been accused of killing 11 children and injuring 181 others when an antibiotic was tested on them during a meningitis epidemic in 1996. The company denies the claims, saying they were victims of the outbreak. The Kano State lawyer told the BBC compensation would be paid to victims, but figures could not yet be disclosed. Barrister Aliyu Umar said money would also be given to a local hospital.

H: Pfizer is accused of murdering 11 children.

If the meaning of H cannot be inferred from the meaning of T, either T contradicts H (contradiction pair) or the truth of H cannot be judged on the basis of T (unknown pair). The first pair bellow is "contradiction pair", whereas the second one is an "unknown pair":

T: Former Peruvian President Alberto Fujimori has been sentenced to 25 years in jail for ordering killings and kidnappings by security forces. At the end of a 15-month trial, judges found him guilty of two death-squad killings of 25 people during the conflict with guerrillas in the 1990s. After being sentenced, Mr Fujimori said he would appeal against the verdict. Human rights group Amnesty International described the verdict as "a milestone in the fight for justice".

H: Alberto Fujimori has been sentenced to life in prison.

T: The amount of water flowing into the stricken Murray River between January and March was the lowest for that quarter in the 117 years that records have been kept. An unprecedented drought has thrown the river system into decline, according to the guardian for the river. "We've had big droughts before and big floods before, but what we didn't have was climate change," said Rob Freeman, the chief executive of the Murray-Darling Basin Authority.

H: The Murray-Darling Basin is in Australia.

[1] `http://pages.cs.aueb.gr/nlp`

[2] See `http://www.nist.gov/tac/2009/RTE/index.html`.

So, this year's challenge was a three-way classification task, but the original two-way task was also preserved. Each team could submit up to three runs per task (three-way or two-way). Moreover, in this year's task the participants should perform at least one ablation test. An ablation test consists of removing one module at a time from a system, and rerunning the system on the test set with the other modules, except the one tested. In this way one can observe the impact of different knowledge resources in textual entailment recognition.

In the following section, we describe our participation in the TAC 2009 RTE track. We used a supervised machine learning algorithm with several similarity measures as features. We also used WordNet (Fellbaum, 1998) to detect synonyms and additional features to measure similarities of grammatical relations obtained by a dependency parser.[3]

3 System overview

Our system uses a Maximum Entropy (ME) (Jaynes, 1957; Good, 1963) classifier[4] to distinguish between the three different categories (namely entailment, contradiction, and unknown) that a T–H pair can be classified in. The classifier is trained with vectors having as features various similarity measures, under the assumption that similarities at various shallow abstractions of the input (e.g., the original sentences, the stems of their words, their POS tags) can be used to recognize textual entailment reasonably well. This approach attempts to improve our previous systems (Malakasiotis and Androutsopoulos, 2007; Galanis and Malakasiotis, 2008) that participated in the 3rd and 4th RTE challenges (Giampiccolo et al., 2007). Based on our experience from previous work (Malakasiotis, 2009) we now try to exploit WordNet in a better way and we also use features that measure the grammatical similarity of T and H.

We employ 9 string similarity measures that are applied to the following 10 pairs of strings, which correspond to 10 different levels of abstraction of T and H. These pairs are:

pair 1: two strings consisting of the *original tokens* of T and H, respectively, with the original order of the tokens maintained;[5]

pair 2: as in the previous case, but now the tokens are replaced by their *stems*;

pair 3: as in the previous case, but now the tokens are replaced by their *part-of-speech* (POS) tags;

pair 4: as in the previous case, but now the tokens are replaced by their *soundex codes*;[6]

pair 5: two strings consisting of only the *nouns* of T and H, as identified by a POS-tagger, with the original order of the nouns maintained;

pair 6: as in the previous case, but now with *nouns replaced by their stems*;

pair 7: as in the previous case, but now with *nouns replaced by their soundex codes*;

pair 8: two strings consisting of only the *verbs* of T and H, as identified by a POS-tagger, with the original order of the verbs maintained;

pair 9: as in the previous case, but now with *verbs replaced by their stems*;

pair 10: as in the previous case, but now with *verbs replaced by their soundex codes*.

A common problem in textual entailment is that T may be much longer than H, which may mislead the string similarity measures. Consider, for example, the following T–H pair where H appears almost verbatim in T, but the length difference yields low similarity.

T: Nigeria's Kano State and US drugs firm Pfizer have agreed to settle a multi-million dollar lawsuit out of court, lawyers for both sides say. Pfizer has been accused of killing 11 children and injuring 181 others when an antibiotic was tested on them during a meningitis epidemic in 1996. The company denies the claims, saying they were victims of the outbreak. The Kano State lawyer told the BBC compensation would be paid to victims, but figures could not yet be disclosed. Barrister Aliyu Umar said money would also be given to a local hospital.

H: Pfizer is accused of murdering 11 children.

To address this problem, when we consider a pair of strings (s_1, s_2), if s_1 is longer than s_2, we also compute the nine values $f_i(s'_1, s_2)$, where f_i $(1 \leq i \leq 9)$ are the string similarity measures, for every s'_1 that is a substring of s_1 of the same length as

[3] We use Stanford University's ME classifier and parser; see http://nlp.stanford.edu/.

[4] We use Stanford University's implementation; see http://nlp.stanford.edu/.

[5] We use Stanford University's tokenizer and POS-tagger, and our own implementation of Porter's stemmer.

[6] Soundex is an algorithm intended to map English names to alphanumeric codes, so that names with the same pronunciations receive the same codes, despite spelling differences; see http://en.wikipedia.org/wiki/Soundex.

s_2. We then locate the s'_1 with the best average similarity to s_2, shown below as s'^*_1:

$$s'^*_1 = \arg\max_{s'_1} \sum_{i=1}^{9} f_i(s'_1, s_2)$$

and we keep the nine $f_i(s'^*_1, s_2)$ values and their average as 10 additional measurements. Similarly, if s_2 is longer than s_1, we keep the nine $f_i(s_1, s'^*_2)$ values and their average. This process is applied only to pairs 1–4; hence, there is a total of 40 additional measurements in each T–H case.

The measurements discussed above provide 130 numeric features that can be used by the induced classifier.[7] To those, we add two Boolean features indicating the existence or absence of negation in T or H, respectively; negation is detected by looking for words like "not", "won't" etc. Finally, we add a length ratio feature, defined as $\frac{\min(L_T, L_H)}{\max(L_T, L_H)}$, where L_T and L_H are the lengths, in tokens, of T and H. Hence, there is a total of 133 available features.

Note that the similarities are measured in terms of tokens, not characters. For instance, the edit distance of T and H is the minimum number of operations needed to transform T to H, where an operation can be an insertion, deletion or substitution of a single token. Moreover, we use high-level POS tags only, i.e., we do not consider the number of nouns, the voice of verbs etc.; this increases the similarity of positive type 3 pairs.

3.1 String similarity measures

We now describe the nine string similarity measures that we use. The reader is reminded that the measures are applied to string pairs (s_1, s_2), where s_1 and s_2 correspond to the ten aforementioned abstractions of T and H, respectively.

Levenshtein distance: This is the minimum number of operations (edit distance) needed to transform one string (in our case, s_1) into the other one (s_2), where an operation is an insertion, deletion, or substitution of a single character. In pairs of strings that contain POS tags and soundex codes, we consider operations that insert, delete, or substitute entire tags, instead of characters.

Jaro-Winkler distance: The Jaro-Winkler distance (Winkler, 1999) is a variation of the Jaro distance (Jaro, 1995), which we describe first. The Jaro distance d_j of s_1 and s_2 is defined as:

$$d_j(s_1, s_2) = \frac{m}{3 \cdot l_1} + \frac{m}{3 \cdot l_2} + \frac{m - t}{3 \cdot m},$$

where l_1 and l_2 are the lengths (in characters) of s_1 and s_2, respectively. The value m is the number of characters of s_1 that match characters of s_2. Two characters from s_1 and s_2, respectively, are considered to match if they are identical and the difference in their positions does not exceed $\frac{\max(l_1, l_2)}{2} - 1$. Finally, to compute t ('transpositions'), we remove from s_1 and s_2 all characters that do not have matching characters in the other string, and we count the number of positions in the resulting two strings that do not contain the same character; t is half that number.

The Jaro-Winkler distance d_w emphasizes prefix similarity between the two strings. It is defined as:

$$d_w(s_1, s_2) = d_j(s_1, s_2) + l \cdot p \cdot [1 - d_j(s_1, s_2)],$$

where l is the length of the longest common prefix of s_1 and s_2, and p is a constant scaling factor that also controls the emphasis placed on prefix similarity. The implementation we used considers prefixes up to 6 characters long, and sets $p = 0.1$.

Again, in pairs of strings (s_1, s_2) that contain POS tags or soundex codes, we apply this measure to the corresponding lists of tags in s_1 and s_2, instead of treating s_1 and s_2 as strings of characters.

Manhattan distance: Also known as City Block distance or L_1, this is defined for any two vectors $\vec{x} = \langle x_1, \ldots, x_n \rangle$ and $\vec{y} = \langle y_1, \ldots, y_n \rangle$ in an n-dimensional vector space as:

$$L_1(\vec{x}, \vec{y}) = \sum_{i=1}^{n} |x_i - y_i|.$$

In our case, n is the number of distinct words (or POS tags or soundex codes) that occur in s_1 and s_2 (in any of the two); and x_i, y_i show how many times each one of these distinct words occurs in s_1 and s_2, respectively.

Euclidean distance: This is defined as follows:

$$L_2(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

In our case, $\vec{x}$ and $\vec{y}$ correspond to s_1 and s_2, respectively, as in the previous measure.

[7] All feature values are normalized in $[-1, 1]$. We use our own implementation of the string similarity measures.

Cosine similarity: The definition follows:

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \cdot \|\vec{y}\|}.$$

In our system $\vec{x}$ and $\vec{y}$ are as above, except that they are binary, i.e., x_i and y_i are 1 or 0, depending on whether or not the corresponding word (or POS tag or soundex code) occurs in s_1 or s_2, respectively.

N-gram distance: This is the same as L_1, but instead of words we use all the (distinct) character n-grams in s_1 and s_2; we used $n = 3$.

Matching coefficient: This is $|X \cap Y|$, where X and Y are the sets of (unique) words (or tags) of s_1 and s_2, respectively; i.e., it counts how many common words s_1 and s_2 have.

Dice coefficient: This is the following quantity; in our case, X and Y are as in the previous measure.

$$\frac{2 \cdot |X \cap Y|}{|X| + |Y|}$$

Jaccard coefficient: This is defined as $\frac{|X \cap Y|}{|X \cup Y|}$; again X and Y are as in the matching coefficient.

3.2 Exploiting WordNet

Using only string similarity measures has the risk of missing true entailment relationships that are due to the existence of synonyms. Therefore, during the calculation of the similarity measures we treat words from T and H that are synonyms according to WordNet (Fellbaum, 1998) as identical.

3.3 Grammatical similarity

The features described so far operate at the lexical level. In this year's participation we added features that operate on the grammatical relations (dependencies) a dependency grammar parser returns for T and H. We use three measures (resulting to a total of 136 features) to calculate similarity at the level of grammatical relations, namely T dependency recall (R_T), T dependency precision (P_T) and their F-measure (F_{R_T,P_T}), defined below:

$$R_T = \frac{|common\ dependencies|}{|T\ dependencies|}$$

$$P_T = \frac{|common\ dependencies|}{|H\ dependencies|}$$

$$F_{R_T,P_T} = \frac{2 \cdot R_T \cdot P_T}{R_T + P_T}$$

As with POS-tags, we use only the highest level of the tags of the grammatical relations, which increases the similarity of positive pairs of T and H. For the same reason, we ignore the directionality of the dependency arcs which we have found to improve the results in our previous work regarding paraphrasing (Malakasiotis, 2009).

4 Official results and discussion

We submitted a total of six runs, three for the three-way classification task and three for the two-way classification task. The three runs were produced in the same way for both tasks. For the first run we trained our system using the first 133 features (namely all the features except those concerning the grammatical similarity) described in section 3. In the second run, the classifier was trained with the first 133 features and R_T. Finally, in the third run used all the 136 features to train the classifier. The runs for the three-way task were also evaluated as two-way runs, simply by merging "contradiction" and "unknown" pairs to "no entailment" pairs.

As training data we used only the development data of this year's RTE challenge. Preliminary experiments indicated that the use of additional data from other challenges (e.g., including training data from past RTE challenges) reduces the predictive power of the classifier. This might be due to differences in the ways the datasets were constructed. Tables 1 and 2 present the results of our runs. Moreover table 3 presents the results after the ablation of WordNet in our 1st run concerning the three-way classification task.

The results indicate that for the two-way classification task the grammatical similarity is more likely to confuse than help the classifier to distinguish between a true and a false entailment T–H pair. This result is a bit surprising but, regarding the fact that this year's texts were much larger than the hypotheses, is understandable. The same does not stand for the three-way results since these features seem to help the classifier. An other interesting and somewhat unexpected result is that after the ablation of WordNet concerning the 1st run for the three-way task the results where much higher. This means that the detection of synonyms does not seem to help the recognition of entailment. This can be again due to the very long texts. A possible improvement could be to try to detect hypernyms instead of synonyms or even to use met-

Two-way runs					
Run 1		Run 2		Run 3	
Accuracy	Average precision	Accuracy	Average precision	Accuracy	Average precision
0.6100	0.5565	0.6017	0.5478	0.5983	0.5416

Table 1: RTE two-way runs.

Three-way runs					
Three-way results					
Run 1		Run 2		Run 3	
Accuracy		Accuracy		Accuracy	
0.5700		0.5750		0.5717	
Two-way results					
Run 1		Run 2		Run 3	
Accuracy	Average precision	Accuracy	Average precision	Accuracy	Average precision
0.6133	0.5315	0.6150	0.5353	0.6117	0.5301

Table 2: RTE three-way runs.

Three-way runs			
Three-way results			
Run 1		Run 1 with WordNet ablated	
Accuracy		Accuracy	
0.5700		0.5967	
Two-way results			
Run 1		Run 1 with WordNet ablated	
Accuracy	Average precision	Accuracy	Average precision
0.6133	0.5315	0.6333	0.5409

Table 3: RTE Run 1 with WordNet ablated.

rics that measure the semantic relatedness between words.

5 Conclusion

We presented AUEB's participation in TAC 2009 RTE track. We attempted to improve the system with which we participated in the 3rd and 4th RTE challenges. Therefore, apart from employing a supervised learning algorithm and string similarity measures, we also exploited WordNet and a dependency parser. Although in the three-way classification task the grammatical similarity features seem to improve the results the same does not hold for the two-way task. This could be due to much longer, compared to the hypotheses, texts. Moreover, the ablation test indicated that detecting synonyms during the calculation of string similarities confuses rather than helps the classifier. This could be again due the large texts.

References

C. Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press.

D. Galanis and P. Malakasiotis. 2008. Aueb at tac 2008. In *Proceedings of Text Analysis Conference (TAC 2008)*, Gaithersburg, Maryland, USA, November.

D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. 2007. The 3rd PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, Czech Republic.

I. J. Good. 1963. Maximum entropy for hypothesis formulation, especially for multidimentional contigency tables. *Annals of Mathem. Statistics*, 34:911–934.

M.A. Jaro. 1995. Probabilistic linkage of large public health data file. *Statistics in Medicine*, 14:491–498.

E. T. Jaynes. 1957. Information theory and statistical mechanics. *Physical Review*, 106:620–630.

P. Malakasiotis and I. Androutsopoulos. 2007. Learning textual entailment using SVMs and string similarity measures. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 42–47, Prague, Czech Republic.

Prodromos Malakasiotis. 2009. Paraphrase recognition using machine learning to combine similarity measures. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pages 27–35, Suntec, Singapore, August. Association for Computational Linguistics.

W.E. Winkler. 1999. The state of record linkage and current research problems. Statistical Research Report RR99/04, US Bureau of the Census, Washington, DC.