STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVI**, Number 2, 2011

DETECTING TEXTUAL ENTAILMENT WITH CONDITIONS ON DIRECTIONAL TEXT RELATEDNESS SCORES

ALPÁR PERINI

Abstract. There are relatively few entailment heuristics that exploit the directional nature of the entailment relation. Our system uses directional methods based on the Corley and Mihalcea formula [3] for expressing the directional relatedness of texts which is then combined with conditions that must hold for the entailment to be true. The condition used as a starting point is that of Tatar et al [10]. Several other conditions have been generated automatically based on the RTE-2009 development dataset using a variant of Genetic Programming. The word relatedness score required by the formula uses not only identity and synonymy, but almost all the WordNet relations. We show the results that we have obtained by participating at the 2009 and 2010 editions of the RTE challenge.

1. Introduction

Recognizing textual entailment is a key task for many natural language processing (NLP) problems. It consists in determining if an entailment relation exists between two texts: the text (T) and the hypothesis (H). The notation $T \rightarrow H$ says that the meaning of H can be inferred from T.

Even though RTE challenges lead to many approaches for finding textual entailment implemented by participating teams, only few authors exploited the directional character of the entailment relation. That is, if $T \rightarrow H$, it is less likely that the reverse $H \rightarrow T$ can also hold [10]. This is because the entailment relation, unlike the equivalence relation, is not symmetric.

The paper is organized into five sections. Section 2 presents background on text relatedness and entailment that is used in our system. Section 3 details the conditions used inside the system. either manually or automatically. Section 4 contains the experimental results that we have obtained using our

Received by the editors: March 29, 20011.

2000 *Mathematics Subject Classification.* 68T50, 03H65.

1998 *CR Categories and Descriptors.* I.2.7 [**Computing Methodologies**]: Artificial Intelligence – *Natural Language Processing*.

Key words and phrases. textual entailment, directional relation, text relatedness, RTE, WordNet.

implementations. Section 5 concludes and discusses possible ways for improvement.

2. Background

We recall some earlier work on expressing relatedness between texts which depends on the order in which the two texts are considered. Then these relatedness scores are used to formulate a directional entailment heuristic.

We have derived in paper [9] the directional text relatedness based on the formula of Corley and Mihalcea [3]. The proposed *text relatedness score* was defined as follows:

$$rel(T,H)_T = \frac{\sum_{pos} \sum_{T_i \in WS^T_{pos}} (maxRel(T_i) \times idf(T_i))}{\sum_{pos} \sum_{T_i \in WS^T_{pos}} idf(T_i)} \tag{1}$$

A mathematically similar formula could be given for $rel(T,H)_H$ which would obviously produce a different score. In (1), $maxRel(T_i)$ was defined as the highest *relatedness* between word T_i and words from H having the same part of speech as T_i. The relatedness between a pair of words was computed using many WordNet relations, most of which were not symmetric. We used the equals, same synset, hypernym, hyponym, entailment, meronym, holonym relations with decreasing weights starting with 1.0. The relatedness score of the words was then the weight of the highest ranked WordNet relation that takes place between them.

After defining the relatedness of two texts, which depends on their order, paper [9] derived a directional entailment condition for texts of approximately equal length derived from the condition in paper [10]:

$$rel(T,H)_H > rel(T,H)_T \tag{2}$$

Now the summary of the steps needed for detecting the entailment relation between two given texts, T and H [8]. One needs to compute the relatedness score with respect to each text, $rel(T,H)_T$ and $rel(T,H)_H$, by applying (1). Then compare the resulting two scores according to (2). If this condition holds, $T \to H$ has a good probability, otherwise the entailment is less likely.

3. Entailment conditions used inside our system

In this section we describe the component of our system, which uses (directional) conditions on relatedness scores for discovering entailment relations.

As mentioned earlier, condition (2) was for texts of about the same length, so we have empirically tuned it for the RTE-2009 development dataset to

account for the difference in the text lengths, obtaining the following more appropriate condition [8]:

$$rel(T,H)_H > rel(T,H)_T + 0.56 \tag{3}$$

In addition to (3), we have experimented with other, more complex conditions for detecting entailment [8]. These conditions were generated automatically using Gene Expression Programming (GEP) [6, 7], a variant a Genetic Programming (GP), of course using the development dataset as reference.

3.1. **GEP for TE.** Since the text relatedness scores that we are working with are in fact numerical values in the range 0 and 1, it made sense to try the power of GP. In GEP an individual is represented by a linear chromosome, which can contain one or more genes, each one composed of a head and a tail. The head can contain both functions, terminals and constants, while the tail can only contain terminals and constants. Although the structure of a gene is linear, there is a nice translation to obtain an expression tree (ET) from it, which can then be evaluated to produce a numeric value.

Since a GEP chromosome can have more genes, we can easily generate conditions of the form $expr_1 < expr_2$ with two genes each representing an expression (tree) and with a subsumed linking function ('smaller than') between them. Let us define the set of functions $F = \{+, -, *, /\}$ and the set of terminals $T = \{rel(T,H)_H, rel(T,H)_T\}$. Each chromosome will contain a small set of random constants. The fitness of an individual is computed by evaluating the condition that it represents on each entry in the development dataset and counting the number of correct classifications. The individuals in the population are subject to all the genetic operators proposed in [6]. The algorithm is stopped when when there is no change in fitness during the last number of generations.

The proposed approach using GEP can be further extended to generate more complex entailment conditions. We have experimented with individuals representing heuristics of the form [8]

$$(exp_1 < exp_2) \tag{4}$$

$$(exp_1 < exp_2)\mathbf{and}(exp_3 < exp_4) \tag{5}$$

and

$$[(exp_1 < exp_2)\mathbf{and}(exp_3 < exp_4)]\mathbf{or}(exp_5 < exp_6), \tag{6}$$

however other structures for the conditions are easily possible. Both types of chromosomes use subsumed linking functions, 'smaller than' to link two expressions into a (sub-)condition and logical functions to form the final condition from the sub-conditions.

3.2. **GEP at Work – The Obtained Heuristics.** After several runs of the proposed GEP algorithm, we have obtained many conditions that performed better for the development set than the manually constructed one [8].

For the simplest template equation in (4), the two best individuals that GEP generated were:

$$rel(T,H)_T < 0.4527 \times rel(T,H)_H^3 \tag{7}$$

and

$$rel(T,H)_T + 1.15 < rel(T,H)_H^2 + rel(T,H)_H \tag{8}$$

For the template equation in (5), the best individual the GEP has obtained is the following:

$$(1.2837 \times rel(T,H)_T + 0.5 < rel(T,H)_H) \textbf{and} (1.5 \times rel(T,H)_T > 0.1586) \tag{9}$$

The three term template condition from (6) found the following best formula:

$$\begin{aligned} &[(rel(T,H)_T > 0.1061) \textbf{and} (rel(T,H)_T < 0.4527 \times rel(T,H)_H^3] \textbf{or} \\ &(\frac{0.3218}{0.3218 - rel(T,H)_T} < \frac{rel(T,H)_T}{rel(T,H)_H - 0.7518}) \end{aligned} \tag{10}$$

4. Experimental Results

We have developed two separate applications, one in C for generating the heuristics with GEP and the other one in Java for recognizing textual entailment using the proposed conditions.

A part of speech tagger was needed in order to distinguish the open class words. We used the Stanford POS tagger implemented in Java [2] for finding the sets of open-class words. For looking up words and word relations, we used WordNet [5], accessed through the Java interface provided by JWordNet [4].

At this point, we worked with all the possible senses for Ti with the given *pos*. Here a possible improvement is to first disambiguate the word and then work only with the resulted synset. The current implementation simplifies the relatedness formula by considering $idf(w)$ to be always 1 and hence the importance of a word w with respect to some documents is neglected.

Our application participated at the RTE-2009 challenge, therefore it was run several times against the development and testing datasets. The results of the accuracies obtained are summarized in table 1 below:

The results show that even though condition (10) performed better than the other conditions for the development set, it turns out that did not scale well for other data, probably because it made use of the particularities of the data too much. Condition (3) scaled the best for the testing data set,

System	*DevSetAcc(%)*	*TestSetAcc(%)*
Run 1 (3)	60.33	61.50
Run 2 (9)	62.83	59.67
Run 3 (10)	64.33	59.67
RTE best	-	73.50
RTE average	-	61.17
RTE worst	-	50.00

TABLE 1. Comparison of RTE-2009 accuracies obtained for development and testing data sets.

obtaining even better results than for the training set. The fact that the accuracies obtained with it did not oscillate much foreshadows that it is a reliable heuristic for deciding entailment between texts.

Our system participated also at the RTE-2010 challenge, with some necessary slight changes, because here the entailment between two texts had to be decided making use of the document set that it was part of. The new component that was introduced was for parsing all the input data given in the particular format and constructing an object hierarchy of it. This made it possible to form hypothesis and text pairs as it was accepted by the earlier system. The system takes into account only these two sentences when deciding on the truth value of the entailment, ignoring the context of the text that they are part of, as it was the case in previous challenges.

The results of the accuracies obtained are summarized in Table 2 below:

System	*Precision(%)*	*Recall(%)*
Run 1 (3)	38.99	41.80
Run 2 (7)	52.38	15.13
Run 3 (8)	61.76	17.78

TABLE 2. Comparison of RTE-2010 precisions and recalls obtained for the test sets.

The precision results show that condition (8) performed better than the other conditions for the test set. Condition (7) and mainly condition (3) did not scale well for newly seen data. However, condition (3) obtained the best recall measure, while the others were significantly worse. This means that if we are interested in discovering as many potential entailments as possible, condition (3) is better, while if we want a greater certainty for the entailment to hold, then (8) is a compromise solution. Overall the results are acceptable

if we take into account that no sentence context information was used for producing the results.

5. Conclusions and Future Work

In this paper we have presented our systems that participated at the 2009 and 2010 RTE Challenges. The system computed the "similarity" between a pair of words using almost all WordNet relations, hence the name of relatedness. The best result we have obtained for the development dataset was 64.33%, while for the testing dataset the accuracy was 61.50%. As far as the ablation testing for run 3 is concerned, the best result obtained was 61.17%. This accuracy is more than 1% better than the official result for run 3.

Finally, there are possible improvements . Firstly, we can use a word sense disambiguation algorithm for finding the exact sense of the word to work with when computing the relatedness scores. Secondly, we can use the inverse document frequency counts for words, obtained either from [1] or from web searches. Thirdly, both the manually and the automatically generated conditions can be further tuned, mainly by creating individual conditions for each entailment task and then deciding on which one to use based on the task annotation of the text pair.

References

1. *TAC 2009 Recognizing Textual Entailment Track development dataset*, 2009.
2. *Stanford POS tagger*, Jun 2010.
3. C. Corley and R. Mihalcea, *Measuring the semantic similarity of texts*, Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment (Ann Arbor, ed.), 2005, pp. 13–18.
4. I. Feinerer, *wordnet: Wordnet interface*, 2008, R package version 0.1-3.
5. C. Fellbaum, *WordNet: An electornic lexical database*, Bradford Books, 1998.
6. C. Ferreira, *Gene expression programming: a new adaptive algorithm for solving problems*, ArXiv Computer Science e-prints (2001).
7. M. Oltean, *Genetic Programming – Automatic Source Code Generation course*, Tech. report, Babes-Bolyai University, 2009.
8. A. Perini, *Detecting textual entailment with conditions on directional text relatedness scores*, The Fifth PASCAL Recognizing Textual Entailment Challenge (NIST, ed.), NIST, 2010, pp. 1–8.
9. A. Perini and D. Tatar, *Textual entailment as a directional relation revisited*, Knowledge Engineering: Principles and Techniques (2009), 69–72.
10. D. Tatar, G. Serban, A. Mihis, and R. Mihalcea, *Textual entailment as a directional relation*, Journal of Research and Practice in Information Technology **41** (2009), no. 1, 17–28.

Babeş-Bolyai University, Department of Computer Science, Cluj-Napoca, Romania

E-mail address: palpar at gmail.com