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 [CM05] 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 [TSMM09]. 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 using our implementations for the RTE-2009 development and testing datasets.

Contents

1 Introduction **2**

2 Background **2**

2.1 Semantic Text Similarity 3

2.2 Entailment as a Directional Relation 3

3 Overview of the DirRelCond System **3**

4 Inside the DirRelCond System – The Conditions **5**

4.1 GEP for TE 5

4.2 GEP at Work – The Obtained Heuristics 6

5 Experimental Results **6**

6 Conclusions and Future Work **8**

Detecting Textual Entailment with Conditions on Directional Text Relatedness Scores

Alpár Perini [*]

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 [TSMM09]. This is because the entailment relation, unlike the equivalence relation, is not symmetric.

The paper is organized into five sections. Section 2 presents some background on textual entailment that is used in our system. Section 3 gives an overview of our DirRelCond system. Section 4 details how the component of our system which uses conditions for entailment was constructed – either manually or automatically. Section 5 contains the experimental results that we have obtained using our implementations. Section 6 concludes and discusses possible ways for improvement.

2 Background

We recall some existing work on expressing similarity between texts [CM05] which depends on the order in which the two texts are considered. Then these similarity scores are used to formulate a directional entailment heuristic in [TSMM09].

[*]Department of Computer Science, "Babes-Bolyai" University of Cluj-Napoca, Romania, `palpar at gmail.com`

2.1 Semantic Text Similarity

The method is based on the similarity of a pair of documents defined differently depending on with respect to which text it is computed [CM05]. Let $sim(T, H)_T$ denote the similarity between texts T and H with respect to T. Then

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

where *pos* involves the set of open-class words (nouns, verbs, adjectives and adverbs) in each text. The set WS^T_{pos} contains the words T_i from text T that are annotated as having the part of speech *pos*. Here $maxSim(T_i)$ denotes the highest similarity between T_i and words from H having the same part of speech as T_i. A similar formula for $sim(T, H)_H$ could be given.

2.2 Entailment as a Directional Relation

Based on this text-to-text similarity metric, Tatar et al [TSMM09] have derived a textual entailment recognition system. They have demonstrated that in the case when $T \rightarrow H$ holds, the following relation will take place:

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

however, the opposite of this statement is not always true, nevertheless it is likely. In [TSL07] a simpler version for the calculus of $sim(T, H)_T$ is used: namely the only case of similarity is the identity (a symmetric relation) and/or the occurrence of a word from a text in the synset of a word in the other text (not symmetric relation).

3 Overview of the DirRelCond System

After having presented the necessary background, in this section we describe our new DirRelCond system for detecting textual entailment.

First we derive the directional text relatedness based on the formula (1) of Corley and Mihalcea. The proposed *text relatedness score* is defined as follows:

$$rel(T,H)_T = \frac{\sum_{pos} \sum_{T_i \in WS^T_{pos}} (maxRel(T_i) \times idf(T_i))}{\sum_{pos} \sum_{T_i \in WS^T_{pos}} idf(T_i)} \tag{3}$$

A mathematically similar formula could be given for $rel(T, H)_H$ which would obviously produce a different score. In (3), $maxRel(T_i)$ is 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 is computed using many WordNet relations, most of which are not symmetric. We propose the following weights for the different WordNet relations in the final *word relatedness score*:

- equals: 1.0;
- same synset: 0.7;
- hypernyms: 0.5;
- hyponyms: 0.3;
- entailment: 0.2;
- meronyms: 0.15;
- holonyms: 0.15;
- either word not found in WordNet: 0.01.

The relatedness score of the words will be 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, the entailment condition corresponding to (2) now becomes (with the remark that the texts involved are of approximately equal length):

$$rel(T,H)_H > rel(T,H)_T \tag{4}$$

The main difference compared to (2) is that the terms used in (2) involve pure similarity between words in computing (1), consisting of identity and synonymy, both being symmetric. On the other hand, our approach uses the relatedness score between words, based on most of the WordNet relations, many of which are not symmetric.

At this stage, after having introduced the necessary formula and condition for entailment, we will describe the steps needed for detecting the entailment relation between two given texts, T and H. We compute the relatedness score with respect to each text, $rel(T,H)_T$ and $rel(T,H)_H$, by applying (3). We then compare the resulting two scores according to (4). If this condition holds, it is likely that we have a true entailment between the text pairs: $T \rightarrow H$, otherwise the entailment is less likely.

An almost identical system to the one presented here, mainly with changes in the weights of the relations and the conditions used, was presented in the paper [PT09].

4 Inside the DirRelCond System – The Conditions

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 (4) 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:

$$rel(T,H)_H > rel(T,H)_T + 0.56 \tag{5}$$

In addition to (5), we have experimented with other, more complex conditions for detecting entailment. These conditions were generated automatically using Gene Expression Programming (GEP) [Fer01, Olt09], a variant a Genetic Programming (GP), of course using the development dataset as reference.

4.1 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. GP can be used for evolving a population of (simple) computer programs. One of the basic examples for its use is symbolic regression, where we need to find the expression of a function that best approximates a set of output values. In the setting used by our system it seems natural to translate the entailment condition to conditions made up of two or more expressions organized into a predefined form. GEP due to its particular nature can best accommodate for this task.

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 $F = \{+, -, \times, /\}$ 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 [Fer01]. 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

$$(exp_1 < exp_2) \wedge (exp_3 < exp_4) \tag{6}$$

and

$$[(exp_1 < exp_2) \wedge (exp_3 < exp_4)] \vee (exp_5 < exp_6), \tag{7}$$

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.

4.2 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. For the second and third run we have used the best formula obtained for a two and the three component template condition respectively.

For the template equation in (6), the best individual that our GEP implementation has obtained is the following:

$$(1.2837 \times rel(T,H)_T + 0.5 < rel(T,H)_H) \wedge (1.5 \times rel(T,H)_T > 0.1586) \tag{8}$$

The three-term template condition from (7) produced the following formula:

$$\begin{gathered}[(rel(T,H)_T > 0.1061) \wedge (rel(T,H)_T < 0.4527 \times rel(T,H)_H^3] \vee \\ \left(\frac{0.3218}{0.3218 - rel(T,H)_T} < \frac{rel(T,H)_T}{rel(T,H)_H - 0.7518}\right)\end{gathered} \tag{9}$$

5 Experimental Results

We have developed two separate application, 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 [sta09] for finding the sets of open-class words. For looking up words and word relations, we used WordNet [Fel98], accessed through the Java interface provided by JWordNet [Fei08].

At this point, we worked with all the possible senses for T_i 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.

System	*DevSetAcc(%)*	*TestSetAcc(%)*
Run 1 (5)	60.33	61.50
Run 2 (8)	62.83	59.67
Run 3 (9)	64.33	59.67
RTE best	-	73.50
RTE average	-	61.17
RTE worst	-	50.00

Table 1: Comparison of RTE-2009 accuracies obtained by our DirRelCond system for development and testing datasets.

The results show that even though condition (9) performed better than the other conditions for the development set, it turns out that it did not scale well for newly seen data, probably because it made use of the particularities of the development data too much. Condition (5) obtained manually scaled the best for the testing dataset, 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, especially for the IR task, where it produced a 76% accuracy.

The last stage of the contest required to perform ablation tests. Basically the only component of the system which could be partly removed without breaking the functionality was the part responsible for deciding the entailment using conditions. Since Run 3 is the most complex condition having three parts (sub-components) and it has performed well on the devset and less well on the testset, it made sense to perform the testing for that.

As already mentioned, run 3 used a condition generated according to the template from (7), having three sub-conditions. If we consider the sub-conditions of the form $expr_i < expr_j$ to be (sub-)components, then we have 3 components, denoted by C_1, C_2 and C_3. For testing we formed some more interesting combinations between these components and generated the entailment verdicts using those. Table 2 details the accuracy values obtained for these combinations. The last column lists the relative accuracies compared to the accuracy obtained with the original system for run 3. Positive values mean better performance, while negative values a weaker performance.

Components	*AblationAcc(%)*	*RelAcc(%)*
C_1	55.00	-4.67
C_2	61.17	+1.50
$C_1 \wedge C_2$	59.50	-0.17
$C_2 \vee C_3$	60.83	+1.16
$C_1 \vee C_3$	55.50	-4.17

Table 2: Comparison of RTE-2009 ablation testing accuracies

After analyzing the ablation testing results, we can say that there were two cases where the system obtained after removing some components (sub-conditions) had performed better by more than 1%. We can see that by using only C_2 in the final condition, the accuracy obtained would have been closer to that obtained with the manually constructed condition. Hence we can say that automatically generated conditions could constitue a promising approach indeed.

6 Conclusions and Future Work

In this paper we have presented our DirRelCond system that participated at the 2009 RTE Challenge. The system computed the "similarity" between a pair of words using almost all WordNet relations, hence the name of relatedness. We have explored new ways of automatically generating conditions for TE using GEP. 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 and closer to our best result obtained.

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 [rte05] 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

[CM05] Courtney Corley and Rada Mihalcea. Measuring the semantic similarity of texts. In Ann Arbor, editor, *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18, 2005.

[Fei08] Ingo Feinerer. *wordnet: WordNet Interface*, 2008. R package version 0.1-3.

[Fel98] Christiane Fellbaum. *WordNet: An Electornic Lexical Database.* Bradford Books, 1998.

[Fer01] Candida Ferreira. Gene expression programming: a new adaptive algorithm for solving problems. *ArXiv Computer Science e-prints*, February 2001.

[Olt09] Mihai Oltean. Genetic Programming – Automatic Source Code Generation course. Technical report, Babes-Bolyai University, 2009.

[PT09] Alpar Perini and Doina Tatar. Textual entailment as a directional relation revisited. *Knowledge Engineering: Principles and Techniques*, pages 69–72, 2009.

[rte05] PASCAL recognizing textual entailment challenge development dataset, 2005.

[sta09] Stanford POS tagger, Mar 2009.

[TSL07] Doina Tatar, Gabriela Serban, and M. Lupea. Text entailment verification with text similarities. In Babes-Bolyai University, editor, *Knowledge Engineering: Principles and Techniques*, pages 33–40. Cluj University Press, 2007.

[TSMM09] Doina Tatar, Gabriela Serban, A. Mihis, and Rada Mihalcea. Textual entailment as a directional relation. *Journal of Research and Practice in Information Technology*, 41(1):17–28, 2009.