Textual Entailment as a Directional Relation

Doina Tătar, Gabriela Şerban , Mihiş Andreea
Department of Computer Science
University "Babes-Bolyai", Cluj-Napoca, Romania
dtatar@cs.ubbcluj.ro, *gabis@cs.ubbcluj.ro*, *mihis@cs.ubbcluj.ro*
and Rada Mihalcea
Department of Computer Science and Engineering
University of North Texas, Texas, USA
rada@cs.unt.edu

Abstract

This paper presents three original methods for solving the problem of textual entailment, methods obtained from an equal number of text-to-text metrics. The first method starts with the directional measure of text-to-text similarity presented in [2], and integrates word sense disambiguation and several heuristics. The second method exploits the relations between "cosine" directional measures of similarity as means to identify textual entailment. Finally, the third method relies on the directional variant of Levenshtein distance between two words. Each "word" in this method is a text with all the words concatenated.

In all these methods the decision about an entailment relation depends on the directional relations established between these measures of similarity. This could be a good start for a later human decision about textual entailment, decision which has to consider the lack of monotonicity of real texts or the effects of the multiple negation.

The methods have been applied to the whole set of text-hypothesis pairs contained in PASCAL RTE-1 development dataset [20]

The corresponding accuracy and statistics are presented for each method.

Keywords

words similarity, texts similarity, Word Sense Disambiguation, text entailment

1 Introduction

The text entailment relation between two texts: T (the text) and H (the hypothesis) represents a fundamental phenomenon of natural language. It is denoted by $T \rightarrow H$ and means that the meaning of H can be inferred from the meaning of T. The recognition of textual entailment is one of the most complex tasks in natural language processing (NLP) and the progress on this task is a key to many applications such as question answering, information extraction, information retrieval, text summarization, and others. For example, a Question Answering (QA) system has to identify texts that entail the expected answer. Given a question, the text entails the expected answer form. Similarly, in Information Retrieval (IR) the concept denoted by a query expression should be entailed from relevant retrieved documents. In multi-document summarization a redundant sentence or expression, to be omitted from the summary, should be entailed from other expressions in the summary. In Information Extraction (IE) entailment holds between different text variants that express the same target relation. In Machine Translation (MT) evaluation a correct translation should be semantically equivalent to the standard translation, and thus both translations have to entail each other. Thus, in a similar way with Word Sense Disambiguation which is recognized as generic tasks, solving textual entailment may consolidate the research on applied semantic inference.

Although the problem is not new, most of the automatic approaches have been proposed only recently within the framework of the Pascal Textual Entailment Challenges RTE-1 (2005), RTE-2 (2006) and RTE-3 (2007). The methods implemented by different teams participating at the RTE events cover domains such as machine learning ([7], [9]), semantic graphs ([9]), logical forms ([13]), theorem proving ([1]) and others. Nonetheless, only few authors exploited the *directional* character of the entailment relation, which means that if $T \rightarrow H$ it is unlikely that the reverse $H \rightarrow T$ also holds. From a logical point of view, the entailment relation is alike to the implication which, contrary to the equivalence, is not symmetric. In this paper we present methods for proving textual entailment using the directional character of this relation. In section 1, we review some directional methods used by the best performing systems participating in the RTE-1 and RTE-2 challenges. In section 2 we show how the classical resolution could benefit from some lexical aspects of the texts T and H in a *lexical refutation* method. In section 3 the directional text similarities introduced in [2] is presented and the text entailment is related with them. In this way, the textual entailment verification is reduced to a comparison of two different similarities between T and H. A system that uses this method is also presented and evaluated. In section 4 three directional cosine measures and a corresponding entailment recognition system are presented. Finally, in section 5 a method which uses a modified Levenshtein distance between texts T and H is presented. Section 6 provides an example of estimation with our three methods. Section 7 discusses conclusions and future work.

2 Textual entailment verification by lexical refutation

It is well known that a linguistic text can be represented by a set of logical formulas, called logic forms [14]. From a logical point of view, proving a textual entailment consists of showing that a logical formula is deductible from a set of other formulas. This is a classical (unfortunately) semidecidable problem in logics. Moreover, few sentences can be accurately translated to logical formulas.

In [15] is proposed a refutation method to solve the problem of establishing if $T \rightarrow H$, a method obtained from the classical resolution refutation method, completing the unification of two atoms with some linguistic considerations. The method is called *lexical refutation* and the modified unification *lexical unification*. We present here *lexical refutation* because in Section 3 we introduce formula (2) and prove it on this base.

In [14], for obtaining the logic forms, each *open-class* word in a sentence (that means: noun, verb, adjective, adverb) is transformed into a logic predicate (atom). The method is applied to texts which are part of speech tagged and syntactically analyzed:

- A predicate is generated for every noun, verb, adjective and adverb (possibly even for prepositions and conjunctions). The name of a predicate is obtained from morpheme the word.
- If the word is a noun, then the corresponding predicate will have as argument a variable, as individual object.
- If the word is a verb, then the corresponding predicate will have as first argument an argument for the event (or action denoted by the verb).
- The arguments of verb predicates are always in the order: event, subject, direct object, indirect object (the condition is not necessary for modified unification).
- If the word is an adjective (adverb) it will introduce a predicate with the same argument as the predicate introduced for modified noun (verb).

The lexical unification method of two atoms proposed in [15] supposes the use of a lexical knowledge base (as, for example, WordNet) where the similarity between two words is quantified. In the algorithm of lexical unification we consider that $sim(p, p')$ between two words p, p' is that obtained, for example, by the Word::similarity interface [12]. This similarity between two words is used to calculate a score for the unifiability of two atoms.

In the algorithm of *lexical unification* [15] the input and the output are:

INPUT: Two atoms $a = p(t_1, ..., t_n)$ and $a' = p'(t'_1, ..., t'_m)$, $n \leq m$, threshold τ, where names p and p' are words in WordNet.

OUTPUT: Decision: The atoms are lexical unifiable with a calculated score W and the unificator is σ, OR they are not unifiable (the score W of unification is less than τ). The score W is the sum of all similarities between $p, t_1, ..., t_n$ and $p', t'_1, ..., t'_m$ during the process of unification.

The other "classical" to "lexical" extensions are:

Definition

Two (disjunctive) clauses c_i and c_j provide by *lexical resolution rule (lr)* the (disjunctive) clause c_k with the score τ, written as

$$c_i, c_j \models_{lr} c_k$$

if $c_i = l \vee c'_i, c_j = \neg l' \vee c'_j$, l and l' are lexical unifiable with the score τ and the unificator σ. The resulting clause is $c_k = \sigma(c'_i) \vee \sigma(c'_j)$.

The following definition is a translation of Robinson's property about a set of disjunctive clauses which are contradictory. As "lexical resolution" is used, we denote this property as "lexical contradictoryness" :

Definition

A set of disjunctive clauses C (obtained from formulas associated to sentences of a text) is lexical contradictory with the score τ if the empty clause [] is obtained from the set of formulas C by repeatedly application of the lexical resolution rule:

$$C \models^*_{lr} [\,]$$

and the sum of all scores of lexical resolution rule applications is τ

The test of relation $T \rightarrow H$ is that the score of refutation (the score of all lexical unifications needed in resolutions) is larger than a threshold τ.

The steps of demonstrating by *lexical refutation* that a text T entails the text H with the score τ consist in: translating T in a set of logical formulas T' and H in H'; considering the set of formulas $T' \cup negH'$, where by $negH'$ we mean the logical negation of all formulas in H'; finding the set C of disjunctive clauses of formulas T' and $negH'$; verifying if the set C is contradictory with the score τ'. If $\tau' \geq \tau$ then the text T entails the text H.

Let us remark that the *lexical refutation* is a directional method: to demonstrate $T \rightarrow H$, the set of clauses is obtained from formulas T' and $negH'$ which is different, of course, from the set of clauses considered if $H \rightarrow T$ is to be demonstrated.

2.1 Some directional method in RTE Challenge

The most notable directional method used in RTE-1 Challenge was that of Glickman [5]. He uses as definition: *T entails H iff P(H |T) > P(H).* The probabilities are calculated on the base of Web. The accuracy of the system is the best for RTE-1 (58,5%).

Another directional method is that of Kouylekov [8], who uses the definition : *T entails H iff there exists a sequence of transformations applied to T such that H is obtained with a total cost below of a certain threshold* . The following transformations are allowed: Insertion: insert a node from the dependency tree of H into the dependency tree of T; Deletion: delete a note from the dependency tree of T; Substitution: change a node in the T into a node of H. Each transformation has a cost and the cost of edit distance between T and H , $ed(T,H)$ is the sum of costs of all applied transformations. The entailment score of a given pair is calculated as: $score(T,H) = \frac{ed(T,H)}{ed(,H)}$ where $ed(,H)$ is the cost of inserting the entire tree H. If this score is bigger than a learned threshold, the relation $T \rightarrow H$ holds. The precision of method is of 56%. Our method in section 5 is even "more directional" : for us, when our edit distance (which is a Levenshtein modified distance) fulfills the relation:

$$ed(T,H) < ed(H,T)$$

then the relation $T \rightarrow H$ holds.

Other teams use a definition which in terms of representation of knowledge as feature structures could be formulated as: *T entails H iff H subsumes T* [3]. Even the method used in [11] is a directional one, as the definition used is: *T entails H iff H is not informative in respect to T*. This last property can be verified for all the methods proposed in the following sections.

3 Method 1: Textual entailment using similarity of texts

A method of establishing the entailment relation could be obtained using a directional measure of similarity between two texts presented in [2]. In this paper, the authors define the similarity between the texts T_i and T_j with respect to T_i as:

$$sim(T_i,T_j)_{T_i}$$

$$= \frac{\sum_{pos}(\sum_{w_k \in WS_{pos}^{T_i}}(maxSim(w_k) \times idf_{w_k}))}{\sum_{pos}\sum_{w_k \in WS_{pos}^{T_i}} idf_{w_k}} \quad (1)$$

Here the sets of open-class words (nouns, verbs, adjective and adverbs) in each text segment are denoted by $WS_{pos}^{T_i}$ pos and $WS_{pos}^{T_j}$. For a word w_k with a given *pos* in T_i, the highest similarity of the words with the same *pos* in the other text T_j is denoted by $maxSim(w_k)$.

Starting with this text-to-text similarity metric, we derive a textual entailment recognition system by applying the *lexical refutation* theory presented above. As the hypothesis H is less informative than the text T, for a TRUE pair the following relation will take place:

$$sim(T,H)_T < sim(T,H)_H \quad (2)$$

This relation can be proven using the *lexical refutation* presented in section 2. A draft is the following: to prove $T \rightarrow H$ it is necessary to prove that the set of formulas $\{T, negH\}$ is lexical contradictory (we denote also by T and $negH$ the sets of disjunctive clauses of T and $negH$). That means empty clause must be obtained from this set of clauses. As $negH$ is the support set of clauses, the clauses in $negH$ must be preferred in refutation. The clauses in $negH$ are used in refutation if the unifications of atoms in H with atoms in T are preferred. So, the following relation holds: *the sum of maximum similarities between atoms of T with atoms of H < the sum of maximum similarities between atoms of H with atoms of T* . As atoms are provided by words, this is exactly the relation (2), with $T_i = T$ and $T_j = H$ and ignoring $idf(w)$.

The criterion obtained from (2) has been applied for the development dataset of RTE-1 with the *path* measure and the obtained accuracy was of 55 % [10].

In [17] it is used a modified version of calculus for $sim(T_i,T_j)_{T_i}$. Namely, the only case of similarity is that of identity (which is a symmetric relation) and/or the occurrence of a word from a text in the synset of a word in other text (which is not a symmetric relation).

Formula 2 is applied to texts disambiguated by CHAD algorithm for word sense disambiguation [16]. So, in the formula denoted by (1), it is selected *pos*=noun, *pos*=verb and it is defined the similarity between two words as 1, if the words are equal or they are situated in the same synset set, and 0 otherwise. In this way are identified (or "aligned" in the terms of [9]) the words that have the same part of speech and either words are identical, or belong to the same synset in WordNet.

This identification is completed with a set of heuristics for recognizing false entailment which occurs because of lack of monotonicity of real texts (COND). (The monotonicity supposes that if a text entails another text, then adding more text to the first, the entailment relation still holds [9]).

Let us denote:

- Named entities in T_1=NP_1 (here we count quantity and time)
- Named entities in T_2=NP_2
- I_c = non-named entities common in T_1 and T_2
- $SYN(T_1)_{T_2}$ ={words non-NE, non common, in T_1, which are nouns or verbs, and are contained in a synset of T_2 } $\cup(NP_1 \cap NP_2) \cup I_c = M_1 \cup (NP_1 \cap NP_2) \cup I_c$
- $SYN(T_2)_{T_1}$ ={words non-NE, non common, in T_2, which are nouns or verbs, and are contained in a synset of T_1} $\cup(NP_1 \cap NP_2) \cup I_c = M_2 \cup (NP_1 \cap NP_2) \cup I_c$
- $C_1 = \mid SYN(T_1)_{T_2} \mid$
- $C_2 = \mid SYN(T_2)_{T_1} \mid$
- $W_{T_1} = NP_1 \cup I_c$
- $W_{T_2} = NP_2 \cup I_c$

The condition for text entailment obtained from (1) and (2) is: $C_1 \leq C_2$ (that means $\mid M_1 \mid \leq \mid M_2 \mid$). Here the relation is $\leq$ (not strict) because of the definition of the sets $SYN(T_1)_{T_2}$ and $SYN(T_2)_{T_1}$.

For our heuristics an important situation is that H contains only named entities and common with T words. In this respect, condition $W_H \subseteq W_T$ is the first verified in the algorithm.

```
if W_H ⊆ W_T /* that means NP_2 ⊆ NP_1
  then
        if T_2 = NP_2 ∪ I_c
          then
            if COND
              then
                  not (T_1 → T_2)
              else
                  T_1 → T_2 (case I)
          else
              if C_1 ≤ C_2
                then
                    T_1 → T_2 (case II)
                else
                    not (T_1 → T_2)
  else
```

not ($T_1 \rightarrow T_2$)

In our system the preprocessing step consists in the POS-tagging text and the named entity recognizing. The disambiguation is realized for calculating the sets $SYN(T_1)_{T_2}$ and $SYN(T_2)_{T_1}$ using CHAD algorithm for WordNet based disambiguation [16]. The application is written in JDK 1.5.0.and uses *HttpUnit* 1.6.2 API in order to search WordNet through the dictionary from [18]. We present the results based on the Pascal RTE-1 Challenge. This dataset contains 800 pairs (T, H), balanced between TRUE and FALSE (and thus a random selection baseline would be evaluated at an accuracy of 50%). These pairs have been collected from different domains (tasks): CD (comparable document), QA (question answering), MT (machine translation), IE (information extraction), RC (reading comprehension) and PP (paraphrase acquisition). The data set is balanced to contain equal numbers of $TRUE$ and $FALSE$. The statistics and accuracy by tasks (CD, IE, IR, MT, PP, QA, RC) is presented in Figure 1 and Figure 2. The CHAIN algorithm [16] is applied with overlap measure.

At RTE-1 challenge the results have been evaluated by accuracy and by average precision (Ap) (confidence weighted score) which is defined as:

$$\sum_i \frac{number-of-correct-ann.-up-to-pair(i)}{i}.$$

The statistics also contain the average precision.

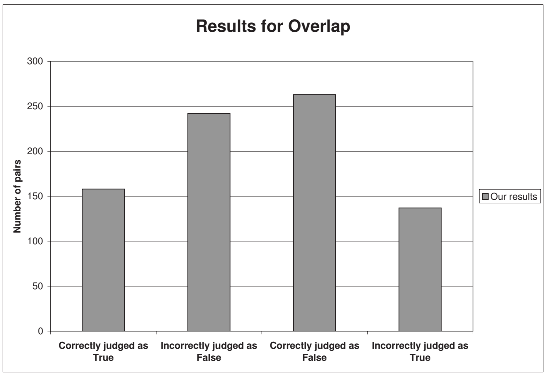

Fig. 1: *Correct and incorrect evaluations for disambiguation method*

4 Method 2: Cosine directional similarity for textual entailment

We define in this section three cosine measure considering the words of $T = t_1, t_2, ...t_m$ and of $H = h_1, h_2, ...h_n$.

- The two vectors for calculating $cos_T(T, H)$ are : $\vec{T} = (1, 1, ...1)$ (a m-dimensional vector) and $\vec{H}$, where $\vec{H}_i = 1$, if t_i is a word in sentence H and $\vec{H}_i = 0$ otherwise.

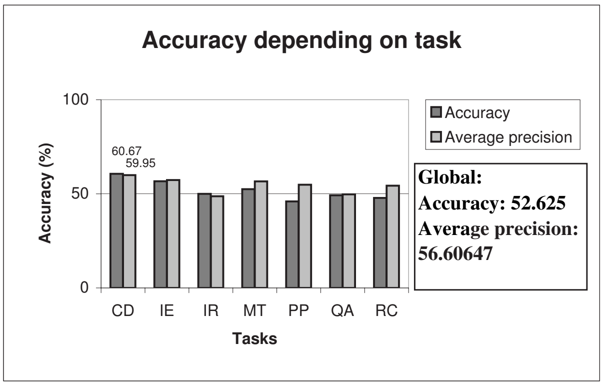

Fig. 2: *Accuracy by tasks for disambiguation method*

- The two vectors for calculating $cos_H(T, H)$ are : $\vec{H} = (1, 1, ...1)$ (a n-dimensional vector) and $\vec{T_i} = 1$, if h_i is a word in sentence T and $\vec{T_i} = 0$ otherwise.
- For $cos_{H\cup T}(T, H)$ the first vector is obtained from the words of T contained in $T \cup H$ and the second, from the words of H contained in $T \cup H$.

Denoting by c the number of common words of T and H, the three measures are: $cos_T(T, H) = \sqrt{\frac{c}{m}}$, $cos_H(T, H) = \sqrt{\frac{c}{n}}$ and $cos_{H\cup T}(T, H) = \sqrt{\frac{4c^2}{(n+c)(m+c)}}$. Relations between them are:

$$cos_H(T, H) \geq cos_{H\cup T}(T, H) \geq cos_T(T, H) \quad (3)$$

considering $m \geq n \geq c$.

Namely, for 94% from the dataset of pairs relation $cos_H(T, H) \geq cos_T(T, H)$ holds, for 97% relation $cos_H(T, H) \geq cos_{H\cup T}(T, H)$ holds and for 76% relation (3) holds. The reason is that $cos_{H\cup T}(T, H) \geq cos_T(T, H)$ only if $c \geq m/3$ and this is fulfilled only for 77% of total set of pairs T, H.

To accomplish the condition: *T entails H iff H is not informative in respect to T* [11], the similarity between T and H calculated with respect to T and to $H \cup T$ must be very closed. Analogously, the similarity between T and H calculated in respect to H and to $H \cup T$ must be very closed. Also, all these three similarities must be bigger than an appropriate threshold. Denoting $cos_T = cos_T(T, H)$, $cos_H = cos_H(T, H)$ and $cos_{HT} = cos_{H\cup T}(T, H)$, the conditions imposed are:

1. $cos_{HT} - cos_T \leq \tau_1$
2. $cos_H - cos_{HT} \leq \tau_2$
3. $max\{cos_T, cos_H, cos_{HT}\} \geq \tau_3$

The threshold founded by a learning method were: $\tau_1 = 0.095, \tau_2 = 0.15$ and $\tau_3 = 0.7$.

Statistics for the accuracy and average precision obtained by tasks, are given in the Figures 3 and 4.

Namely, for CD the accuracy is 73.64 (average precision 74.71), for IE is 61.66 (61.08), for IR is 52.80 (55.38), for MT is 47.5 (42.08), for PP is 58.82 (56.47), for QA is 58.46 (54.41), and for RC is 48.20 (41.65). Let us remark that the best score for CD task is almost a permanent feature for the systems participating at RTE-1.

The accuracy for TRUE pairs is 68.92 and for FALSE pairs is 46.36. The global accuracy is **57.62**.

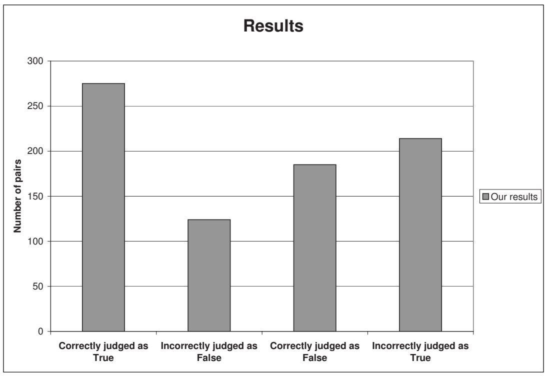

Fig. 3: *Correct and incorrect evaluations for cosine method*

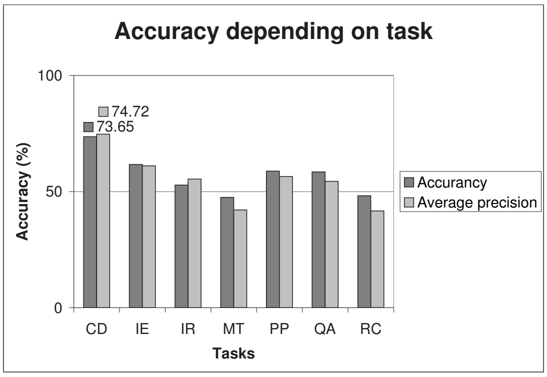

Fig. 4: *Accuracy by tasks for cosine method*

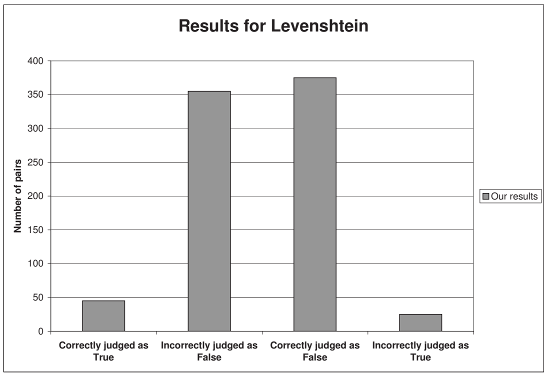

Fig. 5: *Correct and incorrect evaluations for Levenshtein method*

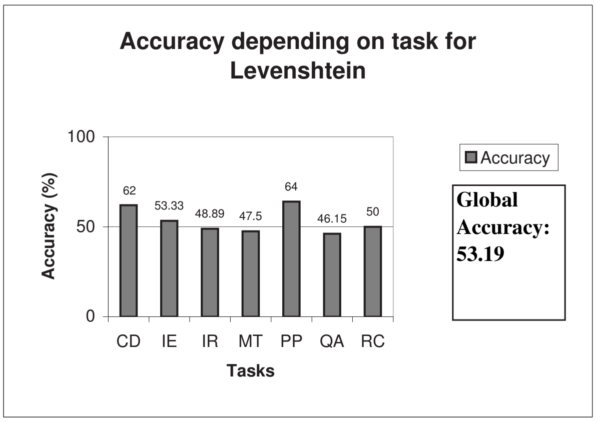

Fig. 6: *Accuracy by tasks for Levenshtein method*

5 Method 3: modified Levenshtein distance for textual entailment verification

Let us consider that for two words w_1 and w_2 the modified Levenshtein distance as calculated by our algorithm is denoted by $LD(w_1, w_2)$. This is defined as the minimal number of transformations (deletions, insertions and substitutions) such that w_1 *is transformed in* w_2 and is, in a way, the quantity of information of w_2 in respect to w_1 . We denote by T_{word} the "word" obtained from the sentence T by considering the empty space as a new letter, and by concatenating all words of T. Analogously is obtained a "word" H_{word}. $LD(T_{word}, H_{word})$ represents the quantity of information of H in respect to T. Let us remark that the modified Levenshtein distance $LD(w_1, w_2)$ is not a distance in usual sense in our algorithm, such that $LD(w_1, w_2) <> LD(w_2, w_1)$.

As T *entails* H *iff* H *is not informative in respect to* T the following relation must hold:

$$LD(T_{word}, H_{word}) < LD(H_{word}, T_{word})$$

We checked the criterion on a set of 800 pairs of RTE-1 development dataset and obtained the results presented in Figure 5 and Figure 6.

The costs of transformations *from the word* w_1 *to the word* w_2 are as following: changecase.cost = 1, insert.cost = 3, remove.cost = 3, substitute.cost = 5, swap.cost = 2.

6 Example

For a better understanding of the way in which our three methods works, the following two examples are analyzed. For the pair with id = 2190, (T=*"Abu Eisa al-Hindi was said to have been involved in a plot to attack Heathrow airport, details of which were allegedly discovered on the computer of Mohammed Naeem Noor Khan, 25, an al-Qa'ida suspect recently arrested in Pakistan."* and H=*"Abu Eisa al-Hindi was recently arrested in Pakistan."*), all the three methods give the correct answer: T does not imply H. For the pair with id = 1861, (T=*"At the welcoming ceremony, Sharon appeared to try to correct the damage from his earlier statements, saying anti-Semitism threatens the Western world, without singling out France."* and H=*"Sharon said anti-Semitism threatened the Western world, but he did not single out France."*), only the last two methods give the correct answer T imply H, but the human can immediately identify it.

7 Conclusions

Establishing entailment relation between two texts using a comparison between some directional similarity measures of that texts could be an easy and elegant method. On the other hand, establishing entailment relation could be the key to many NLP applications. For example, we intend to construct a summarizer based on the entailment relation which is shortly described by the algorithm:

```
Input: Text= {S_1, S_2, ··· S_n}
Output: Summary S.
    S = {S_1}; i = 2
    while i < n do
        if not (S → S_i)
          then
              S := S ∪ {S_i}; i := i + 1
        endif
    endwhile
```

We intend to add to our system a part of semantical heuristics to solve some problems of polarity (negation) and modality, such that this system will be more "computer-aided" and less dependent on human decision.

References

[1] J.Bos, K. Markert: "Recognising Textual Entailment with logical inference", Proceedings of HLT/EMNLP. Vancouver, October 2005, pages 628-635.

[2] C. Corley, R. Mihalcea: "Measuring the semantic similarity of texts", Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, Ann Arbor, June, 2005, pages 13-18.

[3] R. de Salvo Braz, R. Girju, V. Punyakanok, D. Roth, M. Sammons: "An Inference Model for Semantic Entailment in Natural Language" In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, 2005.

[4] I. Dagan, O. Glickman and B. Magnini: The PASCAL Recognising Textual Entailment Challenge, In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, 2005.

[5] O. Glickman, I. Dagan and M. Koppel: "Web Based Probabilistic Textual Entailment", In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, 2005.

[6] A. Haghighi, A. Ng, C. Manning: "Robust textual Inference via graph matching", Proceedings of HLT/EMNLP. Vancouver, October 2005, pages 387-394.

[7] D. Inkpeen,D. Kipp, V. Năstase: "Machine Learning Experiments for textual entailment", In Proc. of the Second PASCAL Challenges on Recognising Textual Entailment, Venice, Italy, 2006.

[8] M. Kouylekov and B. Magnini: "Tree Edit Distance for Recognizing Textual Entailment: Estimating the Cost of Insertion", In Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy, 2006.

[9] B. MacCartney, T. Grenager, M. de Marneffe, D. Cer, C. Manning: "Lerning to recognize features of valid textual entaiments", Proceedings of the HLT Conference of the North American Chapter of the ACL, peges 41-48, New York, June 2006.

[10] R. Mihalcea: private communication, 2007.

[11] C. Monz and M. de Rijke. "Light-Weight Entailment Checking for Computational Semantics", In: P. Blackburn and M. Kohlhase, editors, Proceedings ICoS-3, 2001.

[12] T. Pedersen, S. Patwardhan, and J. Micheelizzi: "Wordnet :: similarity-measuring the relatedness of concepts", Proc. of 5th NAACL, Boston, MA, 2004

[13] R. Raina, A. Ng, C. Manning: "Robust textual inference via learning and abductive reasoning", AAAI, Proceedings of the Twentieth National Conference on AI, 2005.

[14] V. Rus: "Logic form transformation for WordNet glosses and its applications". PhD Thesis, Southern Methodist University, CS and Engineering Department, March 2001.

[15] D. Tătar, M. Frenţiu: "Textual inference by theorem proving and linguistic approach", Studia Universitatis "Babeş- Bolyai", Seria Informatics, 2006, nr 2, pages 31-41.

[16] D. Tătar, G. Şerban, A. Mihiş, M. Lupea and M. Frenţiu : "A chain dictionary method for Word Sense Disambiguation and applications", Proceedings of KEPT2007, Knowledge Engineering Principles and Techniques, Vol I, Presa Universitara Clujeana, p. 41-49.

[17] D. Tătar, G. Şerban, and M. Lupea: "Text entailment verification with text similarities", Proceedings of KEPT2007, Knowledge Engineering Principles and Techniques, Vol I, Presa Universitara Clujeana, p.33-40.

[18] http://wordnet.princeton.edu/perl/webwn

[19] http://ai-nlp.info.uniroma2.it/te/datasets/RTE/RTE2/

[20] http://www.pascal-network.org/Challenges/RTE/