Textual Entailment as a Directional Relation

Doina Tătar, Gabriela Şerban and Andreea Mihiş

Department of Computer Science, University "Babes-Bolyai" Cluj-Napoca, Romania
{dtatar, gabis, mihis}@cs.ubbcluj.ro

Rada Mihalcea

University of North Texas, Computer Science Department
rada@cs.unt.edu

This paper presents three methods for solving the problem of textual entailment, obtained from an equal number of text-to-text similarity metrics. The first method starts with the directional measure of text-to-text similarity presented in Corley and Mihalcea (2005), and integrates word sense disambiguation and several heuristics. The second method exploits the relations between the 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 string consisting of all the words concatenated. In all these methods the decision about an entailment relation depends on the relation established between these measures of similarity. The methods are applied and evaluated on the whole set of text-hypothesis pairs included in the PASCAL RTE-1 development dataset (RTE-1, 2005). The corresponding accuracy and statistics are presented for each method.

Keywords: word similarity, text similarity, Word Sense Disambiguation, Text Entailment
ACM Classification: I.2.7

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 the key to many applications such as Question Answering, Information Extraction, Information Retrieval, Text Summarization, and others. For example, a Question Answering system has to identify texts that entail the expected answer. Given a question, the text entails the expected answer form. Similarly, in Information Retrieval 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 entailment holds between different text variants that express the same target relation. In Machine Translation evaluation a correct translation should be semantically equivalent to the standard translation, and thus both translations have to entail each other. Thus, in a way similar to Word Sense Disambiguation which is recognized as a generic task, solving textual entailment may consolidate the research on applied semantic inference.

Manuscript received: 11 October 2007
Communicating Editor: Vincent Lee

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 (Inkpen *et al*, 2006), semantic graphs (MacCartney, 2006), logical forms (Raina *et al*, 2005), theorem proving (Bos and Markert, 2005) 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 akin to the implication which, unlike the equivalence relation, it is not symmetric. In this paper we present methods for proving textual entailment using the directional character of this relation. 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 2.1 we review some directional methods used by the best performing systems participating in the RTE-1 and RTE-2 challenges. In Section 3 the directional text similarity introduced in Corley and Mihalcea (2005) is presented and the textual entailment is related to 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 that uses a modified Levenshtein distance between texts T and H is presented. Section 6 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 (Rus, 2001). 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, only a few sentences can be accurately translated to logical formulas.

In Tatar and Frentiu (2006) a refutation method is proposed to solve the problem of establishing if $T \rightarrow H$. The method is obtained from the classical resolution refutation method, completing the unification of two atoms with additional linguistic considerations. The method is called *lexical refutation* and the modified unification, *lexical unification*. In Rus (2001), in order to obtain the logic forms, each *open-class* word in a sentence (that means: nouns, verbs, adjectives, adverbs) is transformed into a logic predicate (atom). The method is applied on 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 the morpheme of word.
- If the word is a noun, then the corresponding predicate will have as argument a variable (or a constant).
- 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). Moreover, if the verb is intransitive it will have as arguments two variables (or constants): one for the event and one for the subject argument. If the verb is transitive it will have as arguments three variables (or constants): one for the event, one for the subject and one for the direct complement. If the verb is ditransitive it will have as arguments four variables (or constants): two for the event and the subject and two for the direct complement and the indirect complement.
- The arguments of verb predicates are always in the order: event, subject, direct object, indirect object (the condition is not necessary for the 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).

- If the word is a preposition or a conjunction it will introduce a predicate with the same argument as the modified word.

The *lexical unification* method of two atoms proposed in Tatar and Frentiu (2006) 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 assume that the similarity $sim(p, p')$ between two words p, p', is already available, as for example by using the WordNet:Similarity interface (Pedersen *et al*, 2004). This similarity between two words is then used to calculate a score for the unifiability of the two atoms. In the algorithm of *lexical unification* the input and the output are:

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

OUTPUT: Decision: The atoms are a) lexical unifiable with a calculated score w and the unificator is σ, OR b) they are not unifiable (the score w of unification is less than τ).

The score w is the sum of all the similarities between $p, t_1,\ldots,t_n$ and $p', t'_1,\ldots,t'_m$ during the process of unification. As the score is expected to be large, these similarities are needed to be large.

Let us observe that two terms t_i and t'_j are unifiable in the following two cases.

1. The first case refers to the regular cases in FOPC:
   - terms are equal constants;
   - one term is a variable, the other is a constant;
   - both terms are variables.
2. In the second case, if t_i and t'_j are two different constants, as they are words, then they are unifiable if $sim(t_i, t'_j)$ is large enough. Let us observe that in the method of obtaining logic forms, on which we are based, the arguments of the predicate are only variable or constants.

The similarity $sim(p, p')$ is maximal when p, p' are from the same synset in Wordnet.

The similarity between two words is used to calculate a score for the unifiability of two atoms. The test of unifiability is that the score is larger than a threshold τ.

The other two traditional lexical extensions are:

Definition

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

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

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

Let us denote by $\models^*_{l_r}$ the transitive and reflexive closure of $\models_{l_r}$. The following definition is a translation of Robinson's property about a set of disjunctive clauses which are contradictory. As the *lexical resolution rule* is used, we denote this property as "lexical contradictoriness":

Definition

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

$$C \models^*_{l_r} []$$

and the sum of all scores of lexical resolution rule applications is w.

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

The steps for demonstrating by *lexical refutation* that a text T entails the text H for the threshold τ consist of:

- translating T into 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 w. If $w \geq \tau$ 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 Directional Methods in the RTE-1 Challenge

The most notable directional method used in the RTE-1 challenge was that of Glickman (Glickman *et al*, 2005). They use as definition: *T entails H iff* $P(H \mid T) > P(H)$. The probabilities are calculated on the base of Web. The accuracy of the system was the best for the RTE-1 dataset (58.5%).

Another directional method is that of Kouylekov and Magnini (2006) using 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. When a node is deleted all its children are attached to its parent.
- 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.

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 used a definition which in terms of representation of knowledge as feature structures could be formulated as: *T entails H iff H subsumes T* (de Salvo Braz *et al*, 2005). Even the method used in Monz and de Rijke (2001) is a directional one, as the definition used is: *T entails H iff H is not informative with 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 Corley and Mihalcea (2005). In this paper, the authors define the similarity between the texts T_i and T_j with respect to T_i as:

$$sim(T_i,T_j)_{T_i} = \frac{\sum_{pos}\sum_{w_k \in WS_{pos}^{T_i}}(maxSim(w_k)\cdot idf_{w_k})}{\sum_{pos}\sum_{w_k \in WS_{pos}^{T_i}} idf_{w_k}} \tag{1}$$

Here the sets of open-class words (nouns, verbs, adjectives and adverbs) in each text segment are denoted by $WS_{pos}^{T_i}$ 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. Namely, for the case $T \rightarrow H$ (for a TRUE pair T, H) the following relation will take place:

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

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 lexically 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 as providing a greater score.

So, the following relation holds: *the sum of the maximum similarities between atoms of T with atoms of H < the sum of the 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 to the development dataset of RTE-1 using the *path* measure for an accuracy of 55% (Tatar *et al*, 2007).

In Tatar *et al* (2007a) a modified version of calculus for $sim(T_i,T_j)_{T_i}$ is used. Namely, the only case of similarity is the identity (which is a symmetric relation) and/or the occurrence of a word from a text in the synset of a word in the other text (which is not a symmetric relation).

Formula (2) is applied to texts disambiguated by the CHAD algorithm of word sense disambiguation (Tatar *et al*, 2007b). So, in the formula denoted by (1), it is selected *pos*=noun, *pos*=verb and the similarity between two words is defined as 1, if the words are equal or they belong to the same synset set, and 0 otherwise. In this way are identified (or “aligned”) the words that have the same part of speech and either words are identical, or they belong to the same synset in WordNet.

This identification is completed with a set of heuristics for recognizing false entailment that occurs because of lack of monotonicity of real texts (COND). The monotonicity assumes that if a text entails another text, then adding more text to the first one, the entailment relation still holds (MacCartney *et al*, 2006).

Let us denote:

- Named entities in $T_1 = NP_1$ (here we consider quantity and time)
- Named entities inn $T_2 = NP_2$
- 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 } $\cup (NP_1 \cap NP_2) \cup I_C = M_2 \cup (NP_1 \cap NP_2) \cup I_C$

- $C_1 = | SYN(T_1)_{T_2} |$
- $C_2 = | SYN(T_2)_{T_1} |$
- $W_{T_1} = | NP_1 \cup I_C$
- $W_{T_2} = | NP_2 \cup I_C$

The condition for text entailment obtained from (1) and (2) is: $C_1 \leq C_2$ (that means $| M_1 | \leq | M_2 |$). 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 when H (or T_2) contains only named entities and common with T (or T_i) words. In this case, the condition $W_H \subseteq W_T$ is the first one that is verified in the algorithm.

```
if W_T2 ⊆ W_T1 /* that mean 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 → T_2)
```

In our system the preprocessing step consists of part-of-speech tagging the text and recognizing the named entities. The disambiguation is realized for calculating the sets $SYN(T_1)_{T_2}$ and $SYN(T_2)_{T_1}$ using the CHAD algorithm for WordNet based disambiguation (Tatar *et al*, 2007b). The application is written in JDK 1.5.0. and uses *HttpUnit* 1.6.2 API in order to search WordNet (WordNet, 1998). We present the results based on the Pascal RTE-1 challenge. The 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 is applied with overlap measure.

During RTE-1 challenge the results have been evaluated using accuracy and average precision (confidence weighted score) defined as:

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

The statistics also contain the average precision.

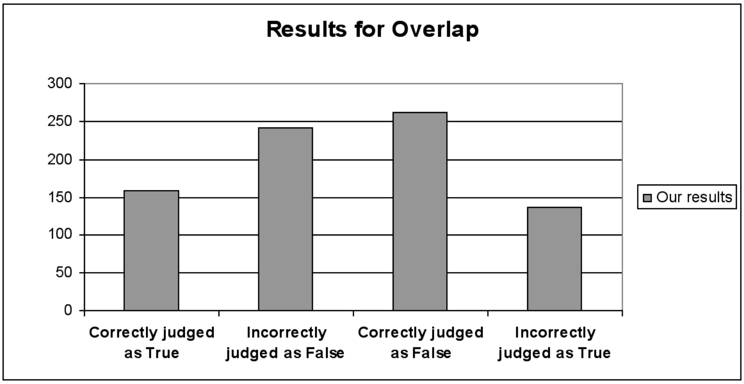

Figure 1: Correct and incorrect evaluations for disambiguation method

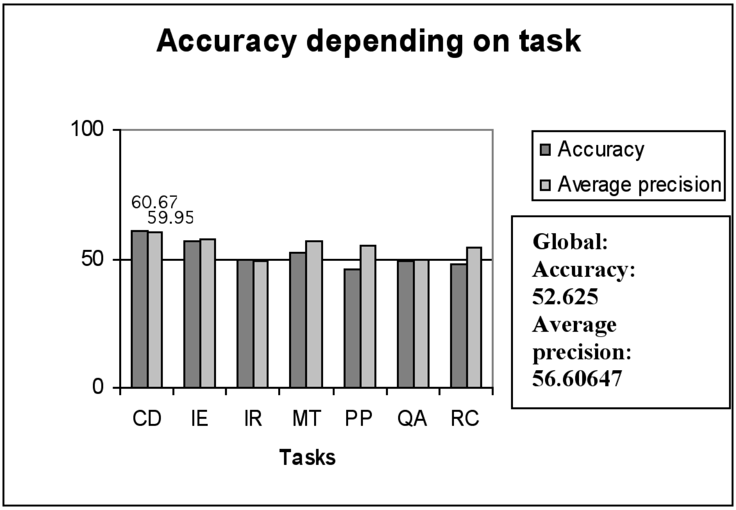

Figure 2: Accuracy by tasks for disambiguation method

4. METHOD 2: COSINE DIRECTIONAL SIMILARITY FOR TEXTUAL ENTAILMENT

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

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

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

with $m \geq n \geq c$.

Namely, for 94% from the dataset of pairs the relation $\cos_H(T,H) \geq \cos_T(T,H)$ holds, for 97% the relation $\cos_H(T,H) \geq \cos_{H\cup T}(T,H)$ holds and for 76% the relation $\cos_H(T,H) \geq \cos_{H\cup T}(T,H) \geq \cos_T(T,H)$ 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 76% of total set of pairs T,H.

To accomplish the condition: *T entails H iff H is not informative in respect to T*, the similarities between T and H calculated with respect to T and to $H \cup T$ must be very closed. Analogously, the similarities 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 larger than an appropriate threshold. Denoting $\cos_T(T,H)$ by $\cos_T$, $\cos_H(T,H)$ by $\cos_H$ and $\cos_{H\cup T}(T,H)$ by $\cos_{HT}$, 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 thresholds found by a learning method are: $\tau_1 = 0.095$, $\tau_2 = 0.15$ and $\tau_3 = 0.7$.

Statistics for the accuracy and the average precision obtained for each task 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),

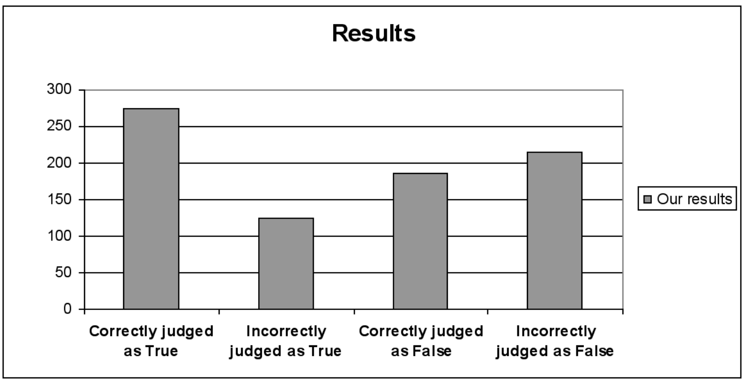

Figure 3: Correct and incorrect evaluations for cosine method

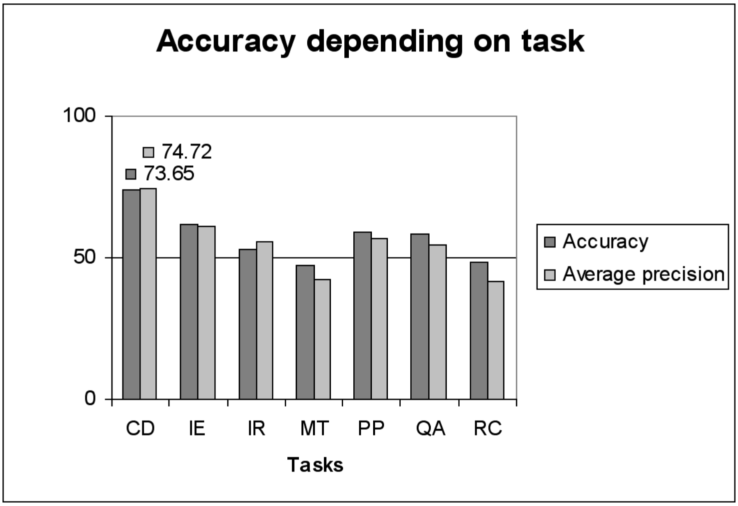

Figure 4: Accuracy by tasks for cosine method

– 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).

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

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, reflecting in a way, the quantity of information of w_2 with 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 the words of T. Analogously a "word" H_{word} is obtained. $LD(T_{word}, H_{word})$ represents the quantity of information of H with respect to T. Let us remark that in our algorithm the modified Levenshtein distance $LD(w_1,w_2)$ is not a distance in the usual sense, such that $LD(w_1,w_2)<>LD(w_2,w_1)$.

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

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

We checked the criterion on the 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 follows:
levenshtein.distance.changecase.cost = 1, levenshtein.distance.insert.cost =3 ,
levenshtein.distance.remove.cost = 3, levenshtein.distance.substitute.cost = 5,
levenshtein.distance.swap.cost = 2.

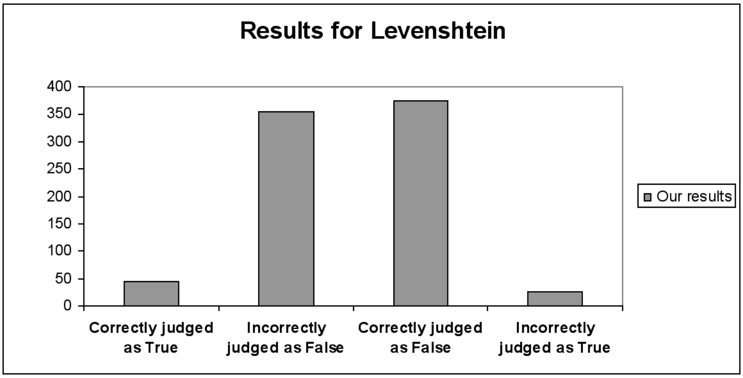

Figure 5: Correct and incorrect evaluations for Levenshtein method

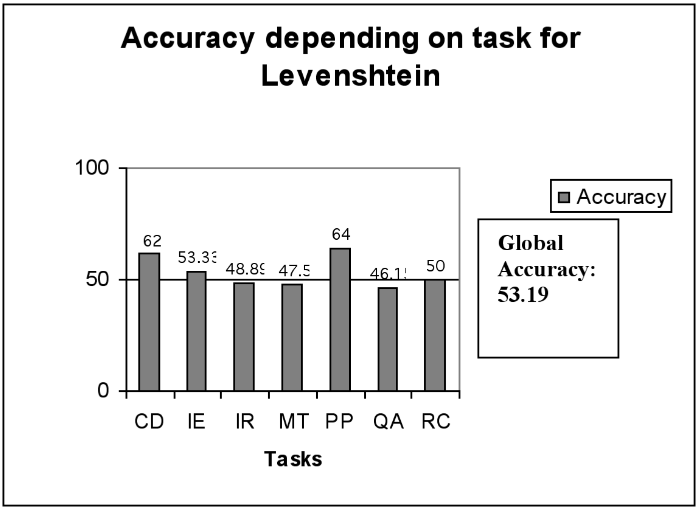

Figure: 6. Accuracy by tasks for Levenshtein method

5.1. CONCLUSIONS

Establishing an entailment relation between two texts using a comparison between directional similarity measures of the texts can be an efficient and elegant method. In future work we plan to build a summarizer based on the entailment relation as briefly described by the following algorithm:

Input: Text= $\{S_1, S_2, \ldots, S n\}$
Output: Summary S

$S = \{S_1\}$;i=2
while $i \leq n$ do
    if not ($S \rightarrow S_i$)
        then
            S:= S $\cup$ $\{S_i\}$;i:= i + 1
    endif
endwhile

On the other hand, there are some issues that impose serious limitations on textual entailment, as for example the lack of monotonicity of natural language texts. Solving these issues could be benefic for textual entailment systems. We intend to augment our system with several semantic heuristics to solve some problems of polarity (negation) and modality.

REFERENCES

BOS, J. and MARKERT, K. (2005): Recognising textual entailment with logical inference, *Proceedings of HLT/EMNLP*, Vancouver, 628-635.

CORLEY, C. and MIHALCEA, R. (2005): Measuring the semantic similarity of texts, *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, Ann Arbor, 13-18.

DE SALVO BRAZ, R., GIRJU, R., PUNYAKANOK, V., ROTH, D. and SAMMONS, M. (2005): An inference model for semantic entailment in natural language. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*. DAGAN, I., GLICKMAN, O. and MAGNINI, B. (eds).

GLICKMAN, O., DAGAN, I. and KOPPEL, M (2005): Web based probabilistic textual entailment. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*. DAGAN, I., GLICKMAN, O. and MAGNINI, B. (eds).

HAGHIGHI, A., NG, A. and MANNING, C. (2005): Robust textual inference via graph matching. *Proceedings of HLT/EMNLP*, Vancouver, 387-394.

INKPEEN, D., KIPP, D. and NASTASE, V. (2006): Machine learning experiments for textual entailment. In *Proceedings of the Second PASCAL Challenges on Recognising Textual Entailment*, Venice, Italy.

KOUYLEKOV, M. and MAGNINI, B. (2006): Tree edit distance for recognizing textual entailment: Estimating the cost of insertion. In *Proceedings of the Second PASCAL Challenges on Recognising Textual Entailment*, Venice, Italy.

MACCARTNEY, B., GRENAGER, T., DE MARNEFFE, M., CER, D. and MANNING, C. (2006): Learning to recognize features of valid textual entailments. *Proceedings of the HLT Conference of the North American Chapter of the ACL*, New York, 41-48.

MONZ, C. and DeRIJKE, M.(2001): Light-weight entailment checking for computational semantics. In *Proceedings ICoS-3*, BLACKBURN, P. and KOLHASE, M. (eds).

PEDERSEN, T., PATWARDHAN, S. and MICHEELIZZI, J. (2004): Wordnet: similarity-measuring the relatedness of concepts. *Proc. of 5th NAACL*, Boston, MA.

RAINA, R., NG, A. and MANNING, C. (2005): Robust textual inference via learning and abductive reasoning, AAAI, *Proceedings of the Twentieth National Conference on AI*.

RUS, V. (2001): Logic form transformation for WordNet glosses and its applications. *Ph.D thesis*. Southern Methodist University, CS and Engineering Department.

TATAR, D. and FRENTIU, M. (2006): Textual inference by theorem proving and linguistic approach. Studia Universitatis “Babeş- Bolyai”, *Informatics*, 2:31-41.

TATAR, D. , SERBAN, G., MIHIS, A. and MIHALCEA, R. (2007): Text entailment as directional relation. In *CALP Workshop at RANLP*, Borovetz, Bulgaria , 53-58. ORASAN, C. and KUEBLER, S. (eds),

TATAR, D., ŞERBAN, G. and LUPEA, M.(2007a): Text entailment verification with text similarities. Proceedings of KEPT2007, Knowledge Engineering Principles and Techniques, Cluj University Press, 33-40.

TATAR, D., ŞERBAN, G., MIHIS, A., LUPEA, M. and FRENTIU, M (2007b): A chain dictionary method for Word Sense Disambiguation and applications. *Proceedings of KEPT2007, Knowledge Engineering Principles and Techniques*, Cluj University Press, 41-49.

WORDNET (1998): http://wordnet.princeton.edu/perl/webwn. Accessed 27-Nov-2008.

RTE1 (2005): http://www.pascal-network.org/Challenges/RTE/ . Accessed 12-Dec-2008.

BIOGRAPHICAL NOTES

Rada Mihalcea

Rada Mihalcea is an associate professor of Computer Science at the University of North Texas. Her research interests are in lexical semantics, graph-based natural language processing, minimally supervised natural language learning, and multilingual natural language processing. She served as the president of the ACL Special Group on the Lexicon (SIGLEX) during 2004–2007. She serves or has served on the editorial boards of the Journals of Computational Linguistics, Language Resources and Evaluations, Natural Language Engineering, Research in Language in Computation, and the Journal of Interesting Negative Results in Natural Language Processing. She is the recipient of a National Science Foundation CAREER award.

Andreea-Diana Mihis is an assistant professor at the Department of Computer Science, Faculty of Mathematics and Computer Science from the Babes-Bolyai University of Cluj-Napoca, Romania, and a PhD student at the same university. She published more than 20 papers, some in prestigious conferences proceedings (ICMI 45 (2006), KEPT 2007, RANLP 2007, NLBD 2008, CiCLing 2008, STEP 2008, SYNASC 2008, ICDIM 2008). Her research interests include formal methods and natural language processing.

Andreea-Diana Mihis

Gabriela Serban is an associate professor at the Department of Computer Science, Faculty of Mathematics and Computer Science from the Babes-Bolyai University of Cluj-Napoca, Romania. She has received the PhD degree in Computer Science in 2003, with the "cum laude" distinction. She published more than 85 papers in prestigious journals and conferences proceedings. Her research interests include artificial intelligence, machine learning, multi-agent systems, programming paradigms.

Gabriela Serban

Doina Tatar is a professor at the Faculty of Mathematics and Computer Science from the Babes-Bolyai University, Cluj-Napoca, Romania. Her teaching and research interests centre round natural language processing, namely semantical and pragmatical aspects. She is also interested in automated theorem proving. She serves as a Program Committee member of the International Conferences: AIMSA, ICCP, SYNASC, KEPT, CALP and as an Editorial Board member of International Journal of Organizational and Collective Intelligence.

Doina Tatar