KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 33–40

TEXT ENTAILMENT VERIFICATION WITH TEXT SIMILARITIES

DOINA TĂTAR(1), GABRIELA ŞERBAN(2), AND MIHAIELA LUPEA(3)

ABSTRACT. This paper presents a new method for recognizing the text entailment obtained from the text-to-text metric introduced in [3] and from the modified resolution introduced in [12]. In [11], using the directional measure of similarity as presented in [3], which measures the semantic similarity of a text T_1 with respect to a text T_2, some conditions of text entailment are established.

In this paper we present a method based on the results presented in [12] and [11], method which supposes the word sense disambiguation of the two texts T_1 and T_2 (text and hypothesis) and adds some appropriate heuristics. The algorithm is applied to a part of the set of pairs (text-hypothesis) contained in PASCAL RTE-2 data [16].

1. Text entailment verification by logical methods

Establishing entailment relationship between two texts is one of the most complex tasks in Natural Language Understanding. Thus, a very important problem in some computational linguistic applications (as question answering, summarization, information retrieval, and others) is to establish if a text *follows* from another text. The progress on this task is the key to many Natural Language Processing applications. Although the problem is not new, most of the automatic approaches have been proposed only recently, in the framework of RTE challenges events. (This year the on-line contest Pascal RTE Challenge is at the third edition.)

Let us denote the entailment relation between a text T_1 and a text T_2 as $T_1 \Rightarrow T_2$. The implemented methods of different teams participating at RTE events cover domains as Machine Learning ([6], [7]), semantic graphs ([7]), logical form ([9]), theorem proving ([2]) and others.

It is well known that a linguistic text can be represented by a set of logical formulas, called logic forms. From a logical point of view, proving a textual entailment consists of showing that a logical formula is deducible from a set of others formulas. This is a classical (semidecidable) problem in logics. Unfortunately, few

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

©2007 Babeş-Bolyai University, Cluj-Napoca

sentences can be accurately translated to logical formulas. This is a reason that "pure" logical methods fail to obtain satisfactory results.

In [12] is proposed a new method to solve the problem of establishing if $T_1 \Rightarrow T_2$, a method obtained from the classical resolution refutation method, completing the unification of two atoms with some linguistic considerations. It is used here the method of obtaining logical forms (in fact, logical formulas) from sentences expressed in natural language proposed by [10] . In this method each *open-class* word in a sentence (that means: noun, verb, adjective, adverb) is transformed in a logic predicate (atom). Unification lexical method of two atoms proposed in [12] 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 by the Word::similarity interface [8], an on-line interface which calculates the similarity between two words using some different similarity measure. The similarity between two words is used to calculate a score for unifiability of two atoms. The test of quality of modified resolution is that the score is larger than a threshold τ.

The steps of demonstrating by resolution (refutation) that a text T_1 entails the text T_2 with the weight τ consist in:

- translating T_1 into a set of logical formulas T_1' and T_2 into T_2';
- considering the set of formulas $T_1' \cup negT_2'$, where by $negT_2'$ we means the logical negation of formulas T_2';
- finding the set C of disjunctive clauses obtained from the set of formulas T_1' and $negT_2'$;
- verifying if the set C is lexical contradictory with the weight τ'. If $\tau' \geq \tau$ then the text T_1 entails the text T_2.

2. Semantic similarity of texts

In [3] the authors introduce a method that combines word-to-word similarity metrics into a text-to-text metric measure, which indicates the semantic similarity of a text T_1 with respect to a text T_2. For a given pair of texts, they start by creating separated sets of open-class words for nouns, verbs, adjective and adverbs.

The authors define in [3] the similarity between the texts T_1 and T_2 with respect to T_1 as:

$$sim(T_1, T_2)_{T_1} = \frac{\sum_{pos}(\sum_{w_k \in WS^{T_1}_{pos}}(maxSim(w_k) \times idf_{w_k}))}{\sum_{pos}\sum_{w_k \in WS^{T_1}_{pos}} idf_{w_k}} \quad (1)$$

Here the sets of open-class words in each text segment are denoted by $WS^{T_1}_{pos}$ and $WS^{T_2}_{pos}$. The highest similarity of a word w_k with a given *pos* in T_1 with the words of the same *pos* in the other text T_2 is denoted by $maxSim(w_k)$.

This measure, which has a value between 0 and 1, is a measure of the directional similarity, in this case computed with respect to T_1. The authors experiment this measure of text similarity using as measure of word similarity that of Wu and Palmer. This similarity metric measures the depth of the two concepts in the WordNet taxonomy, and the depth of the least common subsumer (LCS), and combines these figures into a similarity score: $Sim_{wup} = \frac{2 \times depth(LCS)}{depth(concept1)+depth(concept2)}$.

2.1. **Text entailment verification using similarity of texts.** In this paper we use a simplified definition of similarity of the words. Namely, the single case of similarity is that of identity (which is a symmetric relation) and/or that of the occurrence of a word from a text in the synset of a word in other text (which is not a symmetric relation).

Starting with the measure of text semantic similarity, the textual entailment $T_1 \Rightarrow T_2$ can be derived based on the following theorem, established in the paper [11]. We denoted here by M_{T_1} the set of words from T_1 such that each of them is of maximal similarity with a word in T_2 and by M_{T_2} the set of words of T_2 such that each of them is of maximal similarity with a word in T_1. With these notations the theorem is:

Theorem

$T_1 \Rightarrow T_2$ if the following conditions hold:

$$sim(T_1, T_2)_{T_1} \leq sim(T_2, T_1)_{T_2} \quad (2)$$

$$M_{T_2} \subset M_{T_1} \quad (3)$$

This theorem reduces the verification of entailment relation $T_1 \Rightarrow T_2$ to the calculus of $sim(T_1, T_2)_{T_1}$ and $sim(T_2, T_1)_{T_2}$. The proof is given using the definition of *the demonstration by modified resolution* introduced in [12]. The atom corresponding to the word with a given *pos* in T_1, which has the highest similarity with a word w_k of the same *pos* in the other text T_2 (denoted in (1) by $maxSim(w_k)$), is the most "plausible" atom selected in the modified resolution process. This keeps the quality of the unification in a resolution step high, as this quality depends on the similarity of the two atoms which combine in this step [12].

In order to apply formulas (2) and (3) in our simplified version of similarity of words, we define two sets of words $SYN(T_1)_{T_2}$ and $SYN(T_2)_{T_1}$ as follows:

$SYN(T_1)_{T_2}$ = *the set of nouns in* T_1 *such that they are contained in a synset of disambiguated nouns in* T_2 $\cup$ *the set of nouns in* T_1 *which are contained in* T_2 $\cup$ *the set of verbs in* T_1 *such that they are contained in a synset of disambiguated verbs in* T_2 $\cup$ *the set of verbs in* T_1 *which are contained in* T_2. Analogously is defined $SYN(T_2)_{T_1}$.

The value denoted in (1) as $sim(T_i, T_j)_{T_i}$ is $C_1 = \mid SYN(T_1)_{T_2} \mid$ and the value $sim(T_j, T_i)_{T_j}$ is $C_2 = \mid SYN(T_2)_{T_1} \mid$.

Formulas 2 and 3 in these new forms are verified for texts disambiguated by CHAD algorithm of word sense disambiguation [13]. So, in the formula denoted by 1, we select pos=noun, pos=verb and we define the similarity between two words as 1, if the words are equal or they are situated in the same synset, and as 0 otherwise. In this way we identify (or "align" in the terms of [7]) 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. The false entailment occurs because of lack of monotone character of real texts. Monotonicity supposes that if a a text entails another text, then adding more text to the first, the entailment relation still holds [7].

The heuristics are represented by the bellow condition COND posed in a fixed situation of T_2.

1. $not \in T_1$ and not $not \in T_2$.

In this case the entailment relation does not hold.

For this particular case, $T_2 = NP_2 \cup I_c$, we check if a modal verb is in the following situations:

2. $can \in T_1$ and not $can \in T_2$ (the heuristics shows that "possibility does not entail actuality");

3. $might \in T_1$ and not $might \in T_2$;

4. $should \in T_1$ and not $should \in T_2$;

5. $before \in T_1$ and $after \in T_2$ or $before \in T_2$ and $after \in T_1$;

6. $over \in T_1$ and $under \in T_2$ or $over \in T_2$ and $under \in T_1$.

In all these cases the entailment relation does not hold.

For description of our algorithm, let us make the following notations:

- Named entities in $T_1 = NE_1$ (here we count quantity and time in T_1)
- Named entities in $T_2 = NE_2$ (here we count quantity and time in T_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} $\cup (NE_1 \cap NE_2) \cup I_c = M_1 \cup (NE_1 \cap NE_2) \cup I_c$
- $SYN(T_2)_{T_1}$ ={words non-NE, non common, in T_2, which are nouns or verbs, and are contained in a synset of T_1} $\cup (NE_1 \cap NE_2) \cup I_c = M_2 \cup (NE_1 \cap NE_2) \cup I_c$
- $C_1 = \mid SYN(T_1)_{T_2} \mid$
- $C_2 = \mid SYN(T_2)_{T_1} \mid$
- $W_{T_1} = NE_1 \cup I_c$ (the named entities and the common words from T_1 are the set of words from T_1 such that each of them is of maximal similarity with a word in T_2, such that $W_{T_1} = M_{T_1}$ in (3))
- $W_{T_2} = NE_2 \cup I_c$ (also $= M_{T_2}$)

The conditions for text entailment obtained from 2 and 3 are:

- a) $C_1 \leq C_2$ (that means $\mid M_1 \mid \leq \mid M_2 \mid$)
- b) $W_{T_2} \subset W_{T_1}$ (that means $NE_2 \subset NE_1$)

For our heuristics an important situation is that T_2 contains only named entities and words also contained in T_1 . In this respect, condition b) is verified first.

Algorithm

```
if W_T2 ⊂ W_T1 /* that means NE2 ⊂ NE1
 then
        if T2 = NE2 ∪ Ic
          then
                if COND
                    then
                        not (T1 ⟹ T2)
                    else
                        T1 ⟹ T2 (case I)
                endif
          else
                if C1 ≤ C2
                    then
                        T1 ⟹ T2 (case II)
                    else
                        not (T1 ⟹ T2)
                endif
        endif
 else
        not ( T1 ⟹ T2)
endif
```

For example, if the disambiguated (all nouns are associated with a WordNet synset) texts are:

$$T_1 = w_1\{synset_1\}\, w_2\{synset_2\}\, w_3\{synset_2\}$$

and

$$T_2 = w_4\{synset_2\}\, w_5\{synset_3\}\, w_2\{synset_4\}$$

and if $\{synset_2\} = \{w_2, w_3, w_4\}$ then $\mid SYN(T_1)_{T_2} \mid = \mid \{w_2, w_3, w_2\} \mid = 3$ and $\mid SYN(T_2)_{T_1} \mid = \mid \{w_4, w_2\} \mid = 2$

The conditions 2 and 3 in the above theorem say that, for our example, relation $T_1 \Rightarrow T_2$ does not hold.

2.2. **Implementation and experiments.** The application is written in JDK 1.5.0.and uses *HttpUnit* 1.6.2 API [14]. Written in Java, HttpUnit is a free software that emulates the relevant portions of browser behavior, including form submission, JavaScript, basic http authentication, cookies and automatic page redirection, and allows Java test code to examine returned pages as text, containers of forms, tables, and links [14]. We have used *HttpUnit* in order to search WordNet through the dictionary from [15]. More specifically, *WebConversation*, *WebResponse* and *WebForm* classes from [13] are used. *WebConversation* is used in order to emulate the browser behavior needed to build the test of the web site from [15].

WebResponse class is used in order to obtain the response to a web request from a web server and *WebForm* class is used in order to simulate the submission of a form.

In our system the preprocessing step consists in POS-tagging text and named entity recognizing. The necessary disambiguation for calculating sets $SYN(T_1)_{T_2}$ and $SYN(T_2)_{T_1}$ is realized using our CHAD algorithm of disambiguation, based on WordNet [13].

We present the results obtained when the system is applied to a set of 35 pairs (text-hypothesis) from the data set of Pascal RTE-2 Challenge. The data set is balanced to contain equal numbers of *yes* and *no*. Additionally, we considered a set of 7 pairs corresponding to the cases 1 to 6 in condition COND and to the situation *not* ($W_{T_2} \subset W_{T_1}$), which is not illustrated in this data set. The result was of 25 correct evaluations, which corresponds to an accuracy of 71,4%. Remark that the participants in the first Pascal RTE workshop reported accuracy from 49,5% to 58,6%, and in the second Pascal RTE from 50,8% to 75,3%

For example, the pair text-hypothesis:

< pair id="27" entailment="YES" task="IE" >

< t > Responding to Scheuer's comments in La Repubblica, the prime minister's office said the analysts' allegations, "beyond being false, are also absolutely incompatible with the contents of the conversation between Prime Minister Silvio Berlusconi and U.S. Ambassador to Rome Mel Sembler." < /t >

< h >Mel Sembler represents the U.S.< /h > < /pair>.

has as output of POS-tagger and NER the following:

< t >Responding/V to P1/NNP comments/N in P2/NNP, the P3/NNP office/N said/V the analysts'/N allegations/N, "beyond being/V false, are/V also absolutely incompatible with the contents/N of the conversation/N between P3/NNP P4/NNP P5/NNP and P6/NNP P7/NNP to P8/NNP P9/NNP P10/NNP".< /t >

< h >P9/NNP P10/NNP represents/V the P6/NNP.< /h >

The output of algorithm is "YES" (case II).

As another example,

<pair id="84" entailment="YES" task="IE">

$< t >$Salvadoran reporter Mauricio Pineda, a sound technician for the local canal Doce television station, was shot and killed today in Morazan department in the eastern part of the country.$< /t >$

$< h >$Mauricio Pineda was killed in Morazan.$< /h >$ < /pair>
has the corresponding output

$< t >$P1/NNP reporter/N P2/NNP P3/NNP, a sound/N technician/N for the local canal/N P4/NNP television/N station/N, was shot/V and killed/V today/N in P5/NNP department/N in the eastern part/N of the country/N.$< /t >$

$< h >$P2/NNP P3/NNP was killed/V in P5/NNP.$< /h >$
and the algorithm output is "YES" (case I).

3. Conclusions and further work

There are some issues which impose big limitations to each text entailment system. One of this is the lack of monotonicity of texts in a natural language. The impact of syntactic features is usually positive. We intend to add to our system a part of shallow syntactical analysis and to establish some syntactic heuristics. This will be an advantage especially for recognizing false entailment. For example, at this stage, our system sets a decision "Yes" to the false entailment 1971 from RTE-1:

T_1: *"U.N. officials are dismayed that Aristide killed a conference called by Prime Minister Robert Malval" ;*

T_2: *"Aristide kills Prime Minister Robert Malval".*

An analysis of different objects of verb "kill" in each sentence will recognize the false entailment. Also, the recognition and analysis of if-clauses will reject as false entailment some other examples.

We intend also to use a more complex similarity between the words from a pair text-hypothesis T_1, T_2. For example the lexical chain built using WordNet between a verb from T_1 and a verb from T_2 [1] could indicate, for different thresholds, different relations between these: the smaller values mean closer relationships, 0 being the distance between members of the same synset.

Much work remains in recognition of nonmonotonicity effects, by creating additional heuristics to deal with specific patterns.

References

[1] A. Andreevskaia, Z. Li, S. Bergler: "Can shallow predicate argument structures determine entailment?", Proceedings of the First Pascal Recognizing Textual Entailment Challenge, 2005.

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

[3] 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.

[4] I. Dagan, O. Glickman and B. Magnini: The PASCAL Recognising Textual Entailment Challenge, Proceedings of the PASCAL Work- shop, 2005.

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

[6] 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.

[7] 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.

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

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

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

[11] D. Tătar, C. Corley and R. Mihalcea: "Text entailment and semantic similarity of texts.", accepted at Data Mining and Information Engineering, The New Forest, UK, 18 - 20 June, 2007 .

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

[13] D. Tatar, G. Serban, A. Mihis, M. Lupea, D. Lupsa and M. Frentiu: "Chain algorithm for WSD", submitted to ACL 2007.

[14] http://httpunit.sourceforge.net/, 2006.

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

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

(1)Department of Computer Science, University "Babes-Bolyai" Cluj-Napoca
E-mail address: dtatar@cs.ubbcluj.ro

(2)Department of Computer Science, University "Babes-Bolyai", Cluj-Napoca
E-mail address: gabis@cs.ubbcluj.ro

(3)Department of Computer Science, University "Babes-Bolyai" Cluj-Napoca
E-mail address: lupea@cs.ubbcluj.ro