STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LI**, Number 2, 2006

RECOGNIZING TEXTUAL ENTAILMENT BY THEOREM PROVING APPROACH

DOINA TĂTAR AND MILITON FRENŢIU

Abstract. We present two original methods for recognizing textual inference. First, is a modified resolution method, used in theorem proving, such that some linguistic considerations are introduced in unification of two atoms. Some recent methods of transforming texts in logic forms are used. Second, is based on semantic relations in text, as presented in WordNet. Both methods provide comparable results.

1. Introduction

The recognition of textual inference is one of the most complex task in Natural Language Understanding. Thus, a very important problem in some computational linguistic applications (as Question Answering, summarization, segmentation of discourse, coherence and cohesion of a discourse and others) is to establish if a sentence *follows* from a text. That means in many applications it is important to establish if some sentences which are not existing in text are logicaly implied (can be inferred) by this text. The importance of text inference in computational linguistic is proved by the fact that in TREC (Text REtrieval Conference) conference (http://trec.nist.gov/) and in RTE conference (Recognizing Textual Entailment, http:// www.pascal-network.org/ Challenges/ RTE/) a permanent task is to establish the textual entailment relation. The RTE contest data set includes 1367 English T, H pairs (567 for training stage in learning methods and 800 for test) . Here the task is to determine if the meaning of one text (the entailed hypothesis, H) can be inferred from the meaning of the other text (the entailing text, T).

On the other hand is well known that a linguistic text can be represented by a set of logical formulas, called logic forms. Various method were given for associating a logical formula with a text: [5, 12, 15, 2]. From logical point of view, if each sentence is represented as a formula, proving a textual inference consists

Received by the editors: September 1, 2006.

2000 *Mathematics Subject Classification.* 68T27, 68T30.

1998 *CR Categories and Descriptors.* I.2.3 Deduction and Theorem Proving, I.2.7 Natural Language Processing.

in showing that a logical formula is deducible from a set of others formulas. The problem is a classical (semidecidable) problem. In last years, when text mining is very important in many AI applications, text inference from both point of view, theorem proving and from linguistics perspective, is a very active field of research. In [13] is presented a system participating in the RTE competition, using some world knowledge axioms and a theorem proving tool. The logic method proposed by us in this paper suppose the modification of classical theorem proving task such that it contains a lexical-chain component.

Let us denote entailment relation between a text T and a sentence or a group of sentences H as $T \Rightarrow H$.

In this paper we propose two methods to solve the problem of establishing if $T \Rightarrow H$: first is obtained from the classical resolution refutation method, completing the unification of two atoms with some linguistic considerations (Lexical Resolution Method or LRM). Our method differs of [11] by the fact that it does not need learning stage and it does not need a graph representation and evaluation. The weight (cost) of a deduction is obtained only from the weights (costs) of each resolution steps. At his turn, the cost of a step of resolution is obtained by similarity considerations using some linguistic tools as WordNet [4] and Word::Similarity [8]. No background knowledge [2] is needed.

The second method is based on lexical chains (paths) for entailment spanning the text T and the text H (Lexical-chains Based Method or LBM). A system of rules for construction of lexical rules corresponding to entailment is established. We claim that LRM and LBM produce similar results.

In section 2 we will define our modified unification of two atoms method, our resolution rule and lexical resolution method (LRM).

In section 3 we will describe LBM method and we will propose another definition for text inference based on the cohesion of texts.

2. Text inference as theorem proving.

Consider a knowledge base formed by a set of natural language sentences, K. Let define a set of inferences rules which is sound, in the sense that it derive true new sentences when the initial sentences in K are true. It is a long debate about formalisms to represent knowledge such that above desiderata be fulfilled [15]. We will use here the method proposed by [12] of obtaining logical forms (in fact, logical formulas) from sentences expressed in natural language. In this method each *open* word in a sentence (that means noun, verb, adjective, adverb) is transformed in a logic predicate (atom). We consider, additionally, that the constants are denoted by the names of words they represent (they are real lexical units). For these atoms we propose a new algorithm for unification which modifies the classical Robinson unification algorithm by adding some lexical relaxations.

The semantic information is used in the way we define unification between two atoms, as described in the following section.

2.1. **Unification lexical method for two atoms.** Unification lexical method of two atoms supposes that we have a lexical knowledge base where the similarity between two words is quantified. Such a lexical knowledge base is WordNet [4], a lexical resource which, from its construction in 1998 at Princeton University, is largely used in many linguistic applications. Moreover, some connected resources are constructed (also free) which make use of WordNet easier. For example, Word::similarity is an on-line interface which calculates the similarity between two words using some different similarity measures, all these starting from WordNet facilities [9, 10] It offers the possibility to calculate similarity between two words, two words annotated with POS, or even two words annotated with POS and sense (in WordNet notation). Measures used to calculate similarity could be nine, the most well known are Path lenght, Leacok and Chodorow, Wu and Palmer and Resnik [4]. Of course, a maximal similarity is between words belonging to the same synset (concept).

In the following algorithm we consider that each word of a natural language sentence is transformed in atom as in [12]. See our section 2.3. The classical unification of atoms is replaced by *lexical unification*, which depends on the similarity in the dictionary WordNet. In the following algorithm we consider that $sim(p, p')$ between two words p, p' is that obtained by the Word::similarity interface.

INPUT: Two atoms $a = p(t_1, ..., t_n)$ and $a' = p'(t'_1, ..., t'_m)$, $n \leq m$, threshold τ, threshold for a step τ' . The names p and p' are also words in a lexical knowledge base.

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 steps of the algorithm are:

Step 1. σ = empty substitution, W=0.
Step 2. If $p \equiv p'$ (similarity is maximal) or $sim(p, p') \geq \tau'$
then $W := W + sim(p, p')$; go to Step 3
else Print : " a and a' are not lexical unifiable"; STOP
Step 3. If (for each $t_i, i = 1, ..., n$ exists t'_j in $\{t'_1, ..., t'_m\}$ such that t_i and t'_j are lexical unifiable and the composition of all unificators is σ' OR for each $t'_j, i = 1, ..., m$ exists t_i in $\{t_1, ..., t_n\}$ such that t_i and t'_j are lexical unifiable) , the composition of all unificators is σ', the score is greater than threshold τ
then

Print: " a and a' are lexical unifiable and $\sigma := \sigma$ *composed with* σ'"
else
Print: " a and a' are not lexical unifiable"
STOP

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

1. First one refers to the regular cases in FOPC:
   - terms are equal constants;
   - one is a variable, the other is a constant;
   - both are variables.

2. In the second case, if t_i and t'_j are two different constants, as they are words in KB, then they are unifiable if $sim(t_i, t'_j) \geq \tau'$. In the method of obtaining logic form, on which we are based, the arguments of predicate are only variable or constants.

3. Additionally, 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 unifiability of two atoms. The test in this case is that the score is larger than a threshold τ. The "assumption cost model" presented in [6] uses a similarity measure for some dependency graphs matching. The difference with our method is that they calculate all unificators and choose the best one (which minimizes a given cost). For the resolution method, we need to obtain the empty clause once. The "cost" of resolution is restricted to be low, while the condition of step threshold is applied.

2.2. **Modified resolution or lexical resolution method.** The modified resolution, called also *lexical resolution method*, *LRM*, consists in considering of lexical unification of two atoms as replacing regular unification:

Definition

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

$$c_i, c_j \models_{lexical\,resolution} c_k \quad or \quad , shortly, \;\; c_i, c_j \models_{lr} c_k$$

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

Remark: by disjunctive clause we mean a disjunction of literals (negated or not negated atoms).

The following theorem is a translation of Robinson's theorem of resolution method:

Theorem

A set of disjunctive clauses C (obtained from formulas associated to sentences of a text) is contradictory if the empty clause [] is obtained from the set of formulas C by the modified resolution:

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

Definition

A set of disjunctive clauses C obtained from formulas associated to sentences of a text is contradictory with the weight τ if the empty clause [] is obtained from the set of formulas C by the modified resolution, and the sum of all steps of resolution is τ.

Definition

A set C of clauses which are proved contradictory when modified resolution is used will be denoted as *lexical contradictory*.

Let us resume the steps of demonstrating by lexical resolution method that a text T entails the sentence H with the weight τ, property denoted by $T \Rightarrow_{LRM,\tau} H$:

- Translate T in a set of logical formulas T' and H in H' (as in the following subsection).
- Consider the set of formulas $T' \cup neg(H')$, where by $neg(H')$ we mean the logical negation of formula H'
- Find the set C of disjunctive clauses of the set of formulas T' and $neg(H')$
- Verify if the set C is lexical contradictory with the weight τ. In this case

$$T \Rightarrow_{LRM,\tau} H$$

2.3. **Logical form derivation from sentences.** We will use the method established by [12] which is applied to texts which are part of speech tagged and syntactic analyzed.

The method is the following:

- 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, as individual object. Ex: *person(x2)*.
- 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: one for the event and one for the subject argument. If the verb is transitive it will have as arguments three variables: 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: 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 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).
  Example: *man-made object* is translated as: *object(x1) AND man-made(x1)*
- If the word is a preposition or a conjunction it will introduce a predicate with the same argument as the modified word.

Some transformation rules that create predicates and assign them arguments are presented in [12]. These are obtained from the set of rules of the syntactic analyzer. For example, the rule for introduction of noun predicate is $ART\ NOUN \longrightarrow noun(x_1)$. The rule for introduction of adverb predicate is: $VERB\ ADVERB \longrightarrow verb(e_1, x_1, x_2)\ AND\ adverb(e_1)$.

Let us consider the following example from [13]:

T: John and his son, George, emigrated with Mike, John's uncle, to US in 1969
H: George and his relative, Mike, came to America

The logical form obtained for T is:

$$John(x_1) \wedge son(x_2) \wedge George(x_2) \wedge emigrated(e_1) \wedge Agent(x_1, e_1)$$
$$\wedge Agent(x_2, e_1) \wedge Mike(x_3) \wedge uncle(x_1, x_3) \wedge Location(e_1, x_4)$$
$$\wedge US(x_4) \wedge Time(e_1, x_5) \wedge 1969(x_5)$$

The logical form obtained for H is:

$$George(x_1) \wedge relative(x_2) \wedge Mike(x_2) \wedge came(e_1) \wedge Agent(x_1, e_1)$$
$$\wedge Agent(x_2, e_1) \wedge America(x_3) \wedge Location(e_1, x_3)$$

Applying the unification lexical method for two atoms and modified resolution for the obtained disjunctive clauses, we obtain empty clause, as follows.

First, the set of clauses for $neg(H))$ is formed by only one disjunctive clause:

$$\neg George(x_1) \vee \neg relative(x_2) \vee \neg Mike(x_2) \vee \neg came(e_1) \vee \neg Agent(x_1, e_1)$$
$$\vee \neg Agent(x_2, e_1) \vee \neg America(x_3) \vee \neg Location(e_1, x_2)$$

Then, if we apply modified unification between the following pairs of atoms, the empty clause is obtained:

$relative(x_2), uncle(x_1, x_3)$
$America(x_3), US(x_4),$
$emigrated(e_1), came(e_1).$

The similarities for the pair $relative, uncle$, for the pair $America, US$ and for the pair $emigrated, came$ are calculated with Word::similarity. So $T \Rightarrow_{LRM,\tau} H$ where the weight τ is the sum of these similarities.

Let us remark that in [13] the result is obtained using additionally 6 axioms.

3. Entailment on linguistic bases

In this section we will introduce another definition for entailment between a text T and a sentence H. This definition is based on the concept of lexical paths and on the semantical relations presented on WordNet.

In the huge knowledge base which is WordNet there are many semantic relations which are defined between synsets of nouns, verbs, adverbs and of adjectives. Synsets in WordNet (or concepts) are set of words which are:

a) with the same POS and

b) are similar as meaning (or synonyms).

The most well known semantical relation is the relation *IS-A* between synsets of nouns (or of verbs). The relations *ENTAIL* and *CAUSE-TO* defined only between synsets of verbs, are the most suited for purposes of entailment study.

We will define a *lexical path for entailment* between two words w_1 and w_2, denoted by $LPE(w_1, w_2)$, a path of the form:

$$LPE(w_1, w_2) = c_1 r_1 c_2 r_2 ...... r_{k-1} c_k$$

where w_1 is from the synset c_1, w_2 is from the synset c_k and each relation r_j is a semantical WordNet relation of the form *IS-A* or *ENTAIL* or *CAUSE-TO* between synsets c_j and c_{j+1}. A *lexical path for entailment*, $LPE(w_1, w_2)$, can be described as a regular expression of the form:

$c_1r_1c_2r_2......r_{k-1}c_k \in ((< concept > (IS-A))^*(< concept > (ENTAIL))^* \mid ((< concept > (IS-A))^*(< concept > (CAUSE-TO)^*)^* < concept >$

The relations *IS-A*, *ENTAIL* and *CAUSE-TO* are transitive and no simetric. Thus the paths $LPE(w_1, w_2)$ and all the concepts defined using them have an orientation from w_1 to w_2.

Definition

$T \Rightarrow_{LPE,\tau} H$ if card($\{LPE(w_1, w_2) \mid w_1 \in T,\ w_2 \in H\}$) is greater than a given threshold τ.

A method to construct a path $LPE(w_1, w_2)$ is to apply the following rules:

- From $c_1\ IS-A\ c_2$ and $c_2\ IS-A\ c_3$ it results $c_1\ IS-A\ c_3$
- From $c_1\ IS-A\ c_2$ and $c_2\ ENTAIL\ c_3$ it results $c_1\ ENTAIL\ c_3$
- From $c_1\ ENTAIL\ c_2$ and $c_2\ IS-A\ c_3$ it results $c_1\ ENTAIL\ c_3$
- From $c_1\ ENTAIL\ c_2$ and $c_2\ ENTAIL\ c_3$ it results $c_1\ ENTAIL\ c_3$
- From $c_1\ IS-A\ c_2$ and $c_2\ CAUSE-TO\ c_3$ it results $c_1\ CAUSE-TO\ c_3$
- From $c_1\ CAUSE-TO\ c_2$ and $c_2\ IS-A\ c_3$ it results $c_1\ CAUSE-TO\ c_3$
- From $c_1\ CAUSE-TO\ c_2$ and $c_2\ CAUSE-TO\ c_3$ it results $c_1\ CAUSE-TO\ c_3$
- From $c_1\ CAUSE-TO\ c_2$ and $c_2\ ENTAIL\ c_3$ it results $c_1\ ENTAIL\ c_3$
- From $c_1\ ENTAIL\ c_2$ and $c_2\ CAUSE-TO\ c_3$ it results $c_1\ ENTAIL\ c_3$

We claim that the following theorem holds:

Theorem

For each given threshold τ there exists a threshold τ' such that the relation $T \Rightarrow_{LPE,\tau} H$ holds iff $T \Rightarrow_{LRM,\tau'} H$ holds.

Also, we can introduce another frame for text inference which is very promising to use: the coherence of a text.

Let define a lexical path $LP(w_1, w_2)$ as a path

$$LP(w_1, w_2) = c_1r_1c_2r_2......r_{k-1}c_k$$

were all semantical relations in WordNet are permitted as r_i [3] and c_i are synsets. The semantical relations in WordNet are:

- hypernymy and his reverse hyponymy,
- meronymy and his reverse holonymy,
- entailment, cause-to and reverse of they,
- antonimy.

Definition

The coherence $coh(TE)$ of a text TE is equal to the number of lexical paths which link two different words from text TE.

In the case of entailment of H from T, the coherence of text $T + H$ (the text H and the text T considered as a single text) is larger than the sum of coherences of the separate texts T, H. In other words, we claim that the following theorem holds:

Theorem
$T \Rightarrow H$ iff $coh(T) + coh(H) \leq coh(T + H)$.

4. Conclusions and further work

In this paper we presented two methods for recognizing textual inference: one is from the logic resolution area, using a modified unification algorithm, the second is a pure semantic lexical method and uses the big facilities offered by the huge semantical dictionary WordNet. We consider that the meaning of these methods has common roots: the similarity between two atoms in unification algorithm and the lexical path for entailment are calculated considering semantical relations which exist between concepts (synsets) in WordNet. A study of the relation between τ, τ' is in our attention.

The combined methods in Artificial Intelligence between approaches so different, as Logic and Linguistics, are very largely developed in the last time. The present paper belongs to this category of combined methods.

References

[1] J.Allen: "Natural language understanding", Benjamin/Cummings Publ., 2nd ed., 1995.
[2] J.Bos, K. Markert: "Recognising Textual Entailment with logical inference", Proceedings of HLT/EMNLP. Vancouver, October 2005, pages 628-635.
[3] G. Miller: "WordNet: a lexical database for english", Communications of the ACM, 38(11), pages 39-41 1995
[4] ed. C. Fellbaum: "WordNet: an electronic lexical database", MIT Press, 1998.
[5] S. Harabagiu, D.Moldovan: "A parallel system for Textual Inference", IEEE Transactions parallel and distributed systems, vol 10, no 11, nov 1999 pages 254-270.
[6] A. Haghighi, A. Ng, C. Manning: "Robust textual Inference via graph matching", Proceedings of HLT/EMNLP. Vancouver, October 2005, pages 387-394.
[7] G. Morrill: "Type LogicalGrammar.Categorial Logic of Signs", Kluwer Academic Publishers, 1994.
[8] T. Pedersen, S. Patwardhan, J. Michelizzi: "WordNet::Similarity - Measuring the Relatedness of Concepts", Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04), July 25-29, 2004, San Jose, CA (Intelligent Systems Demonstration)
[9] T. Pedersen, A. Kulkarni: "Identifying Similar Words and Contexts in Natural Language with SenseClusters", Proceedings of the Twentieth National Conference on Artificial Intelligence, Pittsburgh, 2005. (Intelligent Systems Demonstration)

[10] http://www.d.umn.edu/ tpederse/similarity.html.
[11] R. Raina, A. Ng, C. Manning: "Robust textual inference via learning and abductive reasoning", AAAI, 2005, http:www.aaai.org.
[12] V. Rus: "Logic form transformation for WordNet glosses and its applications". PhD Thesis, Southern Methodist University, CS and Engineering Department, March 2001.
[13] M.Tatu, D. Moldovan: "A semantic approach to recognizing Textual Entailment", Proceedings of HLT/EMNLP, Vancouver, October 2005, pg 371-378.
[14] D. Tatar: "Unification Grammars in Natural Language Processing", in "Recent topics in mathematical and computational linguistic", ed. C. Martin-Vide, G. Paun, Editura Academiei, 2000, pg 289-300
[15] A. Thayse (editor): "From natural language processing to logic for expert systems", John Wiley and Sons, 1990.

E-mail address: dtatar@cs.ubbcluj.ro

E-mail address: mfrentiu@cs.ubbcluj.ro

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