Textual Entailment Recognizing by Theorem Proving Approach

Doina Tatar and Militon Frentiu  
Department of Computer Science  
University ”Babes-Bolyai”, Cluj-Napoca, Romania

Abstract  
In this paper we present two original methods for recognizing textual inference. First one is a modified resolution method such that some linguistic considerations are introduced in the unification of two atoms. The approach is possible due to the recent methods of transforming texts in logic formulas. Second one is based on semantic relations in text, as presented in WordNet. Some similarities between these two methods are remarked.

Key words: unification, resolution, textual inference, WordNet.

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 application it is important to establish if some sentences which are not existing in text are "logical" implied (can be inferred) by this text. The importance of text inference in computational linguistic is proved by the existence of RTE conferences (Recognizing Textual Entailment, http://www.pascal-network.org/Challenges/RTE/), with the forth edition this year, where the main task is to establish the textual entailment relation. For RTE-1 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, 11, 14, 2]. From logical point of view, if each sentence is represented as a formula, proving a textual inference consists in showing that a logical formula is deducible from a set of others.
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 [14]. We will use here the method proposed by [11] of obtaining logical forms (in fact, logical formulas) from sentences expressed in natural language. In this method each open class 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. 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 Lexical unification method for two atoms

Lexical unification method of two atoms supposes that we have a lexical knowledge base where the similarity between two words is quantified. A such of lexical knowledge base is WordNet [4], a lexical resource which, from its construction in 1998 at Princeton University, is largely used in many linguistic application. 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 measure, all these starting from WordNet facilities [9], http://www.d.umn.edu/~tpederse/similarity.html. 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 length, Leacock 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 [11]. See our section 2.3. It can be seen there that the terms (the arguments) of an atom are always variables or constants. The classical unification of atoms is replaced by lexical unification, which depends on the similarity in the dictionary WordNet and which relies on this remark. Thus, two terms are unifiable if: they are equal, they are words in the same WordNet synset or their similarity as words is bigger than a threshold.

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(t1, ..., tn) and a′ = p′(t′1, ..., t′m), n ≤ 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 the score W being bigger than the threshold τ) and the unificator is σ; OR they are not unifiable for the threshold τ.

Step 1. σ = empty substitution, W=0.
Step 2. If p = p′ (similarity is maximal, =1) or sim(p, p′) ≥ τ′
then W := W + sim(p, p′) ; go to Step 3
else STOP: "a and a′ are not lexical unifiable”

Step 3. If (for each ti, i = 1, ..., n exists t′ j in {t′1, ..., t′m}) such that t i and t′ j are lexical unifiable with the score W′, W′ > τ′ and the composition of all unificators is σ′ OR for each t′ j, j = 1, ..., m exists ti in {t1, ..., tn} such that t i and t′ j are lexical unifiable with the score W′′, W′′ > τ′ and the composition of all unificators is σ′), and if the new score W (where each W′ is added )is greater than the threshold τ
then
STOP: "a and a' are lexical unifiable and
σ := σ composed with σ"n
else
STOP: "a and a' are not lexical unifiable"

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

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

   In this case the score of lexical unification is 1.

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 \( \text{sim}(t_i, t'_j) \geq \tau' \).

3. Additionally, the similarity \( \text{sim}(p, p') \) is big when \( p, p' \) are from the same synset in Wordnet.

4. As a reviewer pointed up, for lexical unification \( \text{kill}(Oswald,Kennedy) \) is unifiable with \( \text{kill}(Kennedy,Oswald) \). This is true, however, is hard to obtain for the text \( T \) the transcription \( \text{kill}(Oswald,Kennedy) \) and for the hypothesis \( H \, \text{kill}(Kennedy,Oswald) \), when the same tool for the translation of a text in a logical formula is used.

   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 \( \tau \). 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 modified resolution method, we need to obtain the empty clause once. The "cost" of resolution is restricted to be low (the score is high), while the condition of step threshold is applied.

2.2 Lexical resolution rule

The lexical resolution rule \( LR \), 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 score \( W \), written as

\[
\begin{align*}
\text{s.t.} \quad c_i, c_j & \models_{\text{lexical resolution}} c_k \quad \text{or, shortly,} \quad c_i, c_j \models_{lr} c_k \\
\text{if} \quad c_i = l \lor c'_i, \quad c_j = \neg l' \lor c'_j, \quad l \text{ and } l' \text{ are lexical unifiable with the score } W \\
\text{and the unificator } \sigma. \text{ The resulting clause is } c_k = \sigma(c'_i) \lor \sigma(c'_j).
\end{align*}
\]

Remark: by disjunctive clause we mean a disjunction of literals (negated or not negated atoms).
We will call modified resolution method the repeatedly application of lexical resolution rule. So, the modified resolution is the transitive closure of the lexical resolution.

The following definition is a translation of Robinson’s definition for classical resolution method:

Definition
A set of disjunctive clauses $C$ obtained from formulas associated to sentences of a text is lexical contradictory for the threshold $\tau$ if the empty clause $\Box$ is obtained from the set of formulas $C$ by the modified resolution, and the sum of all scores of lexical resolution steps (rules) is bigger than $\tau$:

$$C \models^*_{lr} \Box$$

As in the case of classical resolution, the modified resolution is a problem of search. If we impose in this search problem to choose each time the clauses with the biggest score of lexical resolution, we obtain the empty clause (in the case of a set of lexical contradictory clauses) with the biggest score of derivation.

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

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

$$T \Rightarrow_{M R,\tau} H$$

2.3 Logical form derivation from sentences
We will use the method established by [11] 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. Example: $\text{person}(x2)$. 

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 [11]. These are obtained from the set of rules of the syntactic analyzer. For example, the rule for the introduction of noun predicate is: ART NOUN → noun(x1). The rule for introduction of adverb predicate is: VERB ADVERB → verb(e1, x1, x2) AND adverb(e1).

Let us consider the following example from [12]:

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:

\[ John(x_1) \land son(x_2) \land George(x_2) \land emigrated(e_1) \land Agent(x_1, e_1) \]

\[ \land Agent(x_2, e_1) \land Mike(x_3) \land uncle(x_1, x_3) \land Location(e_1, x_4) \]

\[ \land US(x_4) \land Time(e_1, x_5) \land 1969(x_5) \]

The logical form obtained for H is:

\[ George(x_1) \land relative(x_2) \land Mike(x_2) \land came(e_1) \land Agent(x_1, e_1) \]

\[ \land Agent(x_2, e_1) \land America(x_3) \land Location(e_1, x_3) \]

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

First, the set of clauses for \( \text{neg}(H) \) is formed by only one disjunctive clause:
Then, if we apply modified unification between the following pairs of atoms, the empty clause is obtained:

\[ \neg \text{George}(x_1) \lor \neg \text{relative}(x_2) \lor \neg \text{Mike}(x_2) \lor \neg \text{came}(e_1) \lor \neg \text{Agent}(x_1, e_1) \lor \neg \text{Agent}(x_2, e_1) \lor \neg \text{America}(x_3) \lor \neg \text{Location}(e_1, x_2) \]

The similarities for the pair \( \text{relative}, \text{uncle} \), for the pair \( \text{America}, \text{US} \) and for the pair \( \text{emigrated}, \text{came} \) are calculated with Word::similarity. So \( T \Rightarrow_{\text{MRM}, \tau} H \) where the threshold \( \tau \) must be be smaller then the sum of these similarities.

Let us remark that in [12] 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 \( \text{IS-A} \) between synsets of nouns (or of verbs). The relations \( \text{ENTAIL} \) and \( \text{CAUSE-TO} \) defined only between synsets of verbs, are the most suited for purposes of entailment study.

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

\[ LPE(w_1, w_2) = c_1r_1c_2r_2......r_{k-1}c_k \]

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 \( \text{IS-A} \) or \( \text{ENTAIL} \) or \( \text{CAUSE-TO} \) between synsets \( c_j \) and \( c_{j+1} \). A \textit{lexical path for entailment}, \( LPE(w_1, w_2) \), can be described as a regular expression of the form:

\[ c_1r_1c_2r_2......r_{k-1}c_k \in ((< \text{concept} > (\text{IS-A}))^* (< \text{concept} > (\text{ENTAIL}))^* | ((< \text{concept} > (\text{IS-A}))^* (< \text{concept} > (\text{CAUSE-TO}))^* < \text{concept} > \]

The relations \( \text{IS-A}, \text{ENTAIL} \) and \( \text{CAUSE-TO} \) are transitive and no symetric. Thus the paths \( LPE(w_1, w_2) \) and all the concepts defined using they have an orientation from \( w_1 \) to \( w_2 \).
Definition

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

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 \( c_1 ENTAIL c_3 \)
- From \( c_1 ENTAIL c_2 \) and \( c_2 ENTAIL c_3 \) 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 \( c_1 ENTAIL c_2 \) and \( c_2 ENTAIL c_3 \) it results \( c_1 ENTAIL c_3 \)

We claim that the following theorem holds:

**Theorem**

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

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 \( \tau, \tau' \) 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 Logical Grammar. 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] R. Raina, A. Ng, C. Manning: "Robust textual inference via learning and abductive reasoning", AAAI, 2005, http:www.aaai.org.

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

[12] M. Tatu, D. Moldovan: "A semantic approach to recognizing Textual Entailment", Proceedings of HLT/EMNLP, Vancouver, October 2005, pg 371-378.

[13] 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
[14] A. Thayse (editor): “From natural language processing to logic for expert systems”, John Wiley and Sons, 1990.