

Recognizing Textual Entailment Via Atomic Propositions

Elena Akhmatova and Diego Mollá

Centre for Language Technology, Division of Information and Communication Sciences, Macquarie University, North Ryde, NSW-2109, Australia
{elena, diego}@ics.mq.edu.au

Abstract. This paper describes Macquarie University's Centre for Language Technology contribution to the PASCAL 2005 Recognizing Textual Entailment challenge. Our main aim was to test the practicability of a purely logical approach. For this, atomic propositions were extracted from both the text and the entailment hypothesis and they were expressed in a custom logical notation. The text entails the hypothesis if every proposition of the hypothesis is entailed by some proposition in the text. To extract the propositions and encode them into a logical notation the system uses the output of Link Parser. To detect the independent entailment relations the system relies on the use of Otter and WordNet.

1 Introduction

Despite its study for over two millennia, Natural language is still a complex and somewhat mysterious system which does not stop to surprise us with its variety of phenomena, and which provides scholars with new and interesting tasks to solve. The advent of computers and the recent availability of increasingly large volumes of digitally-stored textual data have provided new opportunities and challenges for current researchers.

The first PASCAL Recognizing Textual Entailment Challenge (Dagan et al. 2005) highlights the relevance of recognizing textual entailment (henceforth RTE) as a core task within the area of Language Technology. The PASCAL challenge consisted of the recognition of textual entailment between coherent sentences T (*text*) and H (*hypothesis*) where T entails H if the meaning of H, as interpreted in the context of T, can be inferred from the meaning of T. In the following examples, the texts of the pairs labelled as A entail their respective hypotheses, whereas the texts of the pairs labelled as B do not entail their hypotheses:

1. **A text.** *Iraqi militants said Sunday they would behead Kim Sun-Il, a 33-year-old translator, within 24 hours unless plans to dispatch thousands of South Korean troops to Iraq were abandoned.*
hypothesis. *Translator was kidnapped in Iraq.*
B text. *Two Turkish engineers and an Afghan translator kidnapped in December were freed Friday.*
hypothesis. *Translator was kidnapped in Iraq.*

2. **A text.** *A The privately owned spacecraft only got about 400 feet into space, according to radar measurements, but it was enough to confirm that it no longer takes a well-heeled government project to organize space travel.*

hypothesis. *Private spaceship launches.*

B text. *The Federal Aviation Administration's Associate Administrator for Commercial Space Transportation (FAA/AST) has given license approval to Scaled Composites of Mojave, California, permitting the firm to expand flight testing of SpaceShipOne – a privately-financed rocket plane to carry passengers to suborbital altitude.*

hypothesis. *Private spaceship launches.*

RTE has been recognized as a principal task in various Language Technology areas, including Question Answering (QA), Information Retrieval (IR), Information Extraction (IE) and (multi-) document summarisation. Even the task of recognizing paraphrases can be reduced to a RTE task, since if X entails Y and Y entails X then X and Y paraphrase each other. Within the area of QA, to give an example of the relevance of textual entailment, given the question *Who killed Kennedy?*, the text *the assassination of Kennedy by Oswald* entails the expected answer *Oswald killed Kennedy*, even though it does not match the wording of the question fully. It would be advisable, therefore, to solve the general problem of text entailment and to apply the acquired solution to any applications that might need it.

A difficulty in the recognition of textual entailment is the fact that information can be expressed in a great variety of forms. Just to give an example, *X wrote Y* and *X is an author of Y* are paraphrases of each other. But the most important difficulty of textual entailment is the frequent use of word knowledge and common sense axioms to draw inferences. For example, *X's new novel Y appeared in the bookstores* expresses the idea that *X is an author of Y* and consequently *X wrote Y*. The task of textual entailment can only be truly solved via a process of fully understanding the text and the hypothesis. Still, we believe that it is possible to recognize textual entailment to a level of success that improves the related language technologies listed above.

The structure of this paper is as follows. Section 2 focuses on the various sources of entailment in natural language. Section 3 proposes a classification of entailments based on the types of tools required for the entailment task. The tools and resources that could be used in an automatic algorithm for entailment recognition are discussed in Section 4. Section 5 describes the architecture of our system and presents the results of the system performance on the PASCAL RTE test data. Section 6 contains a brief comparison of the system with respect to other participants in the RTE challenge. Finally, Section 7 concludes the paper.

2 Sources of Entailment

In this section we introduce a simple entailment classification according to the possible sources of entailment. There are three main sources of entailment, *syntactic information*, *semantic information*, and *logical information*.

2.1 Syntax

Some forms of entailment are derived from specific syntactic transformations. Natural language allows us to convey information compactly, to hide a part of the information, or alternatively, to give more details about the same fact and even emphasize some particular parts of the message. For example, let us suppose that we want to describe the fact that a new house was built. The following sentences could be used, depending on the main intention of our message:

1. *The builders have built the house.*
2. *The new house has been built by builders.*
3. *A new house has been built.*
4. *It is a new house that has been built.*
5. *It is the builders who have built a new house.*
6. *The building of a new house has just been finished.*
7. *The building of a new house by the builders has just been finished.*
8. *The builders managed to build the new house in time.*
9. *The builders finished building the new house in time.*
10. *The builders had to hurry a lot while building a new house.*
11. *The builders had to hurry. They had to build the house fast.*
12. *A house that has been recently built by the builders is very beautiful.*
13. *A house recently built by the builders is very beautiful.*
14. *The desired construction, namely a new house, has been built.*

As illustrated in these examples, the usual way to communicate who made an action is to use an active form. This is shown, for example, in Sentence 1. If it is not important or even not known who has done the action, a passive form such as Sentence 3 can be used. On the contrary, to stress who built the house (say, if one wonders whether this new house has been built by an owner himself, or by the builders), then a cleft sentence like Sentence 5 can be used. Sometimes information is provided in a more compact form, as in Sentence 6. In this case there are two facts in one sentence. It is supposed here that the reader knew beforehand about the construction of a new house in the district and wanted to get some information about an action as a whole.

As one could see, even if two sentences may express the same information they may give a slightly different connotation. Also, it depends on the kind of information that is already known by the reader, what is the most important to him. We may also distinguish between the text genres under consideration. An article in a newspaper or a book summary will compact information to a higher degree than in an essay or the book content itself. All of this will result in differences between the grammatical constructions used.

Some of the above sentences transmit more information, some less, but all of them entail that a new house was built. A general rule of entailment is that if a sentence expresses two nuggets of information, A and B, then the sentence entails both A and B. Thus, sentences containing more information entail the less informative sentences. As we will see below, we will exploit this central idea in our approach to the PASCAL RTE challenge.

For RTE the awareness of such syntactical diversity of the language will help to recognize that, for example, *The building of a new house has just been finished* entails *A new house has been built*.

2.2 Semantics

Another source of textual entailment is the actual meanings of the words. Not all kinds of past studies in lexical semantics are useful for our purposes. For example, take the traditional classification of nouns between proper and common nouns, or the classification between count and uncount nouns (Quirk et al. 1985, Page 245). They all help to characterize the meaning of a noun. Similarly the traditional classification between dynamic verbs and stative verbs (Quirk et al. 1985, Page 201) help us give a semantic description of a verb. But these are largely syntactically driven categorizations. There are syntactic constructions in the language that take into account this kind of knowledge. The division between proper and common nouns gives us the rules for the usage of articles (in the languages where there are articles), for example. Usually proper nouns are used without an article, though the usage of an article gives us some additional information sometimes. Thus, A Dr. Smith = Some Dr. Smith. *The Robinsons* is a family of Robinsons. *The Netherlands* is a country name, containing the plural. Similarly, dynamic verbs allow the use of progressive forms *He is running*, but stative verbs become unacceptable (or generate marked readings) when used in the progressive *He is knowing the truth*.

More useful for us are the semantic relations existing between words. This type of semantic information shows us the place of the word within a hierarchy of words linked by relations such as hyponymy/hypernymy, synonymy, antonymy, and plain entailment. For example, one can state that the meanings of two words are equal to each other if the words belong to the same synonym group, or that the meaning of one word entails the meaning of all of its hypernyms. For example, *The man saw a poodle* entails *The man saw a dog*, as dog is a hypernym of *poodle*.

2.3 Logic and Knowledge

We have left the logic source of textual entailment to the end of our list though this is the most intuitive one. One sentence entails another sentence if the logical concept of the first implies the logical concept of the second. For example, from the sentence *Russian president Vladimir Putin visited US* one can entail that *Vladimir Putin is a president of Russia* and *Vladimir Putin visited US*. As people have common sense and can use world knowledge we would easily deduce also that Vladimir Putin exists, he is human, he is a resident of Russia, and provided that one knows that it is a constitutional rule that only a person over 35 can become a president in Russia, one can also deduce that Putin is older than 35 years old. Also, if the sentence was found in a news article, we would deduce that it is most probably told about an official visit of president Putin to US, and probably many other things. Consequently, linguistic expressions of all these concepts are entailed from the same sentence.

The source of these entailments would not come through syntactic or lexical analysis this time, though these two types of linguistic information might play an auxiliary role in the process. Instead, the source of these entailments is knowledge representation and reasoning. An example of knowledge representation is a knowledge base that holds lexical axioms such as:

president(X) :- human(X), resident(X), over_age_limits(X).

This knowledge base would be complemented with a tool for reasoning that could work with this type of information.

In the work presented in this paper we use limited world knowledge. In particular we use only information that can be extracted from a lexical resource, plus a few general axioms. But a central idea in our work is the general entailment principle that a sentence entails every piece of information that it conveys or a conjunction of them.

In practice, textual entailment is the combination of syntax, semantics, and logic. This can be seen with the example *Peter tracked down and killed the man*. This sentence is a conjunction of two pieces of information, and therefore it entails *Peter killed the man*. A simple syntactic transformation allows the sentence to entail *The man was killed*. Furthermore, there is a cause-effect relation between *kill* and *die*, and therefore the sentence entails *The man was dead*. Finally, common sense tells us that, modulo very exceptional circumstances, someone who is dead becomes dead forever. Therefore, the sentence entails *The man is dead*.

3 Classification of Entailment

With respect to the tools to be used to find an entailment relation one can distinguish between three types of entailment:

- **Lexico-syntactic entailments:** Entailments that could be detected with the help of syntactic and lexical knowledge only. In other words, the hypothesis is just a lexico-syntactic variant of the text sentence.
- **Descriptive entailments:** Entailments of this group are characterized by the substitution of entire descriptions or definitions with a shorter expression.
- **Knowledge-based entailments:** Entailments that would need some extra knowledge, possibly from some entailment database. Lexical resources and syntax play an auxiliary role only.

3.1 Lexico-Syntactic Entailment

Lexico-syntactic entailment is a type of entailment where the hypothesis is a lexico-syntactic variant of the text. That means that the only tools required to prove the entailment relation are those concerned with the extraction of syntactic structures of the text and hypothesis, plus a lexical database. Examples of this type of entailment are:

text. *A Union Pacific freight train hit five people.*

hypothesis. *A Union Pacific freight train struck five people.*

text. *Satomi Mitarai died of blood loss.*

hypothesis. *Satomi Mitarai bled to death.*

The lexical correspondence might be fairly complicated. In the first example the correspondence is word to word, but this is not always the case. For example, a transitive predicate with its object could correspond to an intransitive predicate, *to bleed = to lose blood*, *to dine = to have dinner*, etc. A noun phrase could be expanded or bunched up in a hypothesis. *a dead man = a man who was killed*, *a writing pen = a pen for writing*, *a dinner cake = a cake prepared for dinner*. These complex correspondences between words add a lot of complications to the RTE task.

There might be a verb ellipsis, as in the following example, where only common sense helps one to assume that a rock group performs during a concert rather than attend it:

text. *Phish disbands after a final concert in Vermont on August 15.*

hypothesis. *Rock band Phish holds final concert in Vermont.*

In contrast to paraphrases we do not have to know this transformation in advance but we need a tool to estimate how probable the change is: we need to find an *entailment score*.

Vanderwende et al. (2005) have shown that up to 49% of all the entailment pairs in the RTE development set belong to this group of entailments. These figures were obtained using an *ideal* parser and an *ideal* lexical database. In particular, 37% of the test items of their evaluation can be handled by syntax, and 49% of the test items can be handled by syntax plus a general purpose thesaurus.

3.2 Descriptive Entailment

Entailments of this group are characterized by the fact that a definition or description is substituted with a term which is equivalent in meaning or more generic. See the description of the notions of *opponents* and *discrimination* in the examples of this section. Besides that, some compression techniques could be used to make it possible to convey as much information as possible in brief sentences. This is done not only via syntactic transformations (e.g. nominalizations) but also via the substitution of words. Consequently, knowledge about the syntactic structure of the sentence alone would not be sufficient to recognize all instances of this type of entailment.

For example:

text. *Israeli Prime Minister Ariel Sharon threatened to dismiss Cabinet ministers who don't support his plan to withdraw from the Gaza Strip.*

hypothesis. *Israeli Prime Minister Ariel Sharon threatened to fire cabinet opponents of his Gaza withdrawal plan.*

The following syntactic compression has been made in the above entailment example:

- Generalization: *Gaza Strip* → *Gaza*
- Nominalization: *plan to withdraw* → *withdrawal plan*
- Lexical substitution: *to dismiss* → *to fire*
- Definition substitution: *ministers who do not support X's plan* → *opponents*.

The above definition substitution is difficult to detect automatically given that it would most likely not appear in standard lexical knowledge bases.

Another example:

text. *The country's largest private employer, Wal-Mart Stores Inc., is being sued by a number of its female employees who claim they were kept out of jobs in management because they are women.*

hypothesis. *Wal-Mart is sued for sexual discrimination.*

The following transformations have been made:

- Lexical substitution: *Wal-Mart Stores Inc.* → *Wal-Mart*.
- Definition substitution: *to be sued by a number of its female employees who claim they were kept out of jobs in management because they are women* → *to be sued for sexual discrimination*.

Again, the definition substitution would be difficult to detect automatically.

3.3 Knowledge-Based Entailment

Finally, there are entailment pairs where common sense background knowledge is needed for their detection. We call them knowledge-based entailments.

text. *Researchers at the Harvard School of Public Health say that people who drink coffee may be doing a lot more than keeping themselves awake — this kind of consumption apparently also can help reduce the risk of diseases.*

hypothesis. *Coffee drinking has health benefits.*

text. *Eating lots of foods that are a good source of fibre may keep your blood glucose from rising too fast after you eat.*

hypothesis. *Fibre improves blood sugar control.*

text. *Mexico City has a very bad pollution problem because the mountains around the city act as walls and block in dust and smog.*

hypothesis. *Poor air circulation out of the mountain-walled Mexico City aggravates pollution.*

This type of entailment is much harder than the other two for the simple reason that currently there are no knowledge bases containing all the common-sense knowledge required, and even if there were any it is not obvious how to find the required information among a sea of unrelated information — see, for example, Mahesh et al.'s (1996) study of the applicability of a well-known attempt to provide a common-sense knowledge base.

One of the important questions that needs to be addressed is what kind of entailment will prove to be useful in real applications. From the point of view of QA the lexico-syntactic and the descriptive entailments most probably will bring an improvement to the performance of a QA system. They are also more accessible to computational methods than the third, more complex type of entailment. Therefore, in our approach we have focused on the first two and left the third type (knowledge-based entailment) for future work.

4 Tools

This section we introduce the tools used in the current system for textual entailment recognition.

4.1 Parser

Given that syntax plays a role in text entailment, we decided not to experiment with bag-of-words approaches and used syntactic information instead. The output of the parser provides us with this syntactic information. For our system we used the Link Grammar Parser (Sleator and Temperley 1991) because it is a robust parser that outputs a ranked list of parse variants (not just the favourite parse). The grammar provided with this parser covers a wide range of sentence structures, and various independent evaluations (e.g. Sutcliffe et al. 1996, Mollá and Hutchinson 2003) show that its accuracy is comparable to that of other wide-coverage parsers. Also the code is freely available, and has been implemented in the C programming language with a C API.

The issue about the availability of multiple parse variants is important because, implicitly or explicitly, parsers try to guess the right parse among a (possibly large) list of possible parses for syntactically ambiguous sentences. Syntactic disambiguation is one of the most important and difficult tasks for a parser, since a sentence may contain hundreds of alternative parses, and context external to the scope of the sentence is often required. Our plan is to allow all the possible inferences resulting from the combined parses of a sentence. Thus, the sentence *The man saw a girl with a telescope* would entail the hypotheses that *A girl has a telescope* as well as *The man has a telescope*, according to the two possible readings of the sentence.

Although our current implementation only uses the first parse returned by Link Parser, a parser that returns all possible parses allows further extensions of our method.

Probably, eventually a formal notion of *text coherence* (or *grammaticality*) should be introduced for textual entailment such that all the coherent interpretations of the text should be used for the task rather than one final parse given by a specific parser.

An additional and important use of the syntactic structures returned by a parser is the construction of the logical form of the sentence. The logical form is required if we want to use logical entailment as a means to find textual entailment, as we do in our system.

4.2 Lexical Database

As mentioned in Section 2.2, hierarchical information about words and word concepts is valuable for the task of entailment recognition. The most popular lexical resource for English is WordNet (Miller 1995). It contains hierarchical information about the concepts expressed by nouns, verbs, adjective and adverbs.

For the concept *table#n#2* (i.e. the second word sense of the noun *table*) with gloss *a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs; "it was a sturdy table"*, for example, WordNet provides the following information:

If *X is a table*, then *X is a piece of furniture*, *X is furnishings*, *X is an artefact*, *X is a physical object* and *X is an entity*.

If *X is a booth*, *X is a breakfast table* or *X is a desk*, then *X is a table*.

Also, WordNet provides information about meronymy (has-part) relationships, derivational information, and synonymy information. For verbs, WordNet presents also relations of causation and entailment. All of this information is useful for the task of RTE. Hyponymy and synonymy are especially indicated to detect entailment. To be more precise, proposition P_1 entails proposition P_2 if P_2 is more general than P_1 . One can say that P_2 is more general than P_1 if all of the concepts in P_2 are either synonyms or they are more general (hypernyms) in P_1 . To compare concepts one can use therefore the hypernym and synset information from WordNet.

The meaning comparison at the word level is a well-defined task that has been approached by many researchers. WordNet relatedness and similarity measures (see, for example, the review provided by Budanitsky and Hirst 2001) might be helpful tools to perform such a meaning comparison. The currently available measures have been developed for applications quite different from the entailment recognition. Still, some redevelopment or adjustment might be used successfully for the RTE task. Inspired in these measures, we have defined a custom-made measure that approximates the degree of entailment between two arbitrary words.

4.3 Logic Prover

Given that entailment is a logical relationship, it is only natural to try and find the logical form of the text and the hypothesis and use a logical prover to test if the text entails the hypothesis. The classical theory of computational semantics suggests first-order Predicate Logic or extensions thereof as a tool for the representation of the sentence meaning (see, for example, Jurafsky and Martin 2000, Chapter 14). Various formalisms have been devised, some of which include methods for the construction of the logical forms (what is called the *semantic interpretation* or *semantic analysis* of the sentence). Given that the comparison of logical forms would eventually be done through automated deduction systems (logic provers), it is important to use a logical form notation that can be converted into the format of the chosen automatic prover.

In addition to the logical forms of the text and hypothesis, the logic prover needs to have access to world knowledge encoded in form of axioms. To avoid flooding the prover with a large set of axioms that would not help detect the entailment, our system uses information extracted from WordNet based on the actual words of the sentences.

The process of finding the logical forms and appropriate world-knowledge axioms is potentially difficult, and the computing complexity of the logical proof could be time-consuming. For these reasons most of the systems participating in PASCAL tried methods that avoided the use of logical forms. We decided to try this natural method to assess its feasibility for the RTE task. In particular, our current system uses the theorem prover Otter (Kalman 2001), which is available on the WWW. We will show the usage of Otter using a very simple example.

text. *A boy bought a desk.*

hypothesis. *A boy bought a table.*

Axiom extracted from WordNet:

X is a desk → *X is a table*

Input for Otter:

exists x exists y exists e (*boy*(x) & *bought*(e , x , y) & *desk*(y)).

all x (*desk*(x) → *table*(x)).

-(exists x , y , e (*boy*(x) & *bought*(e , x , y) & *table*(y))).

Having been given the above input, Otter is capable to prove that the hypothesis is entailed from the text.

5 The Proposed System

The central idea of our approach to RTE is to exploit the basic principle of logical entailment: *A and entails A*. In our approach, the entailment between the text and the hypothesis sentences is detected by comparing the *atomic propositions* found in both sentences. By an atomic proposition we mean a minimal declarative statement (or a small idea) that has truth-conditions (is either true T or false F) and whose truth or falsity does not depend on the truth or falsity of any other proposition. For example, given the sentence *Coffee boosts energy and provides health benefits*, the propositions are *Coffee boosts energy* and *Coffee provides health benefits*. Thus, the meaning of a sentence is represented as the set of atomic propositions contained in it. One has to compare the propositions in order to compare the sentences.

To implement the idea we have used Link Parser (Sleator and Temperley 1991) version 4.1a to obtain a syntactic structure of the sentence. The output of the parser was used to extract the atomic propositions of the sentence. These in turn were converted into logical formulae that served as an input for a logical prover, Otter (Kalman 2001) version 3.3 in our case.

The architecture of the system is presented in Figure 1. The system performs the same analysis for the text and for the hypothesis. In both cases there is a process of parsing and a process of proposition extraction. After the propositions

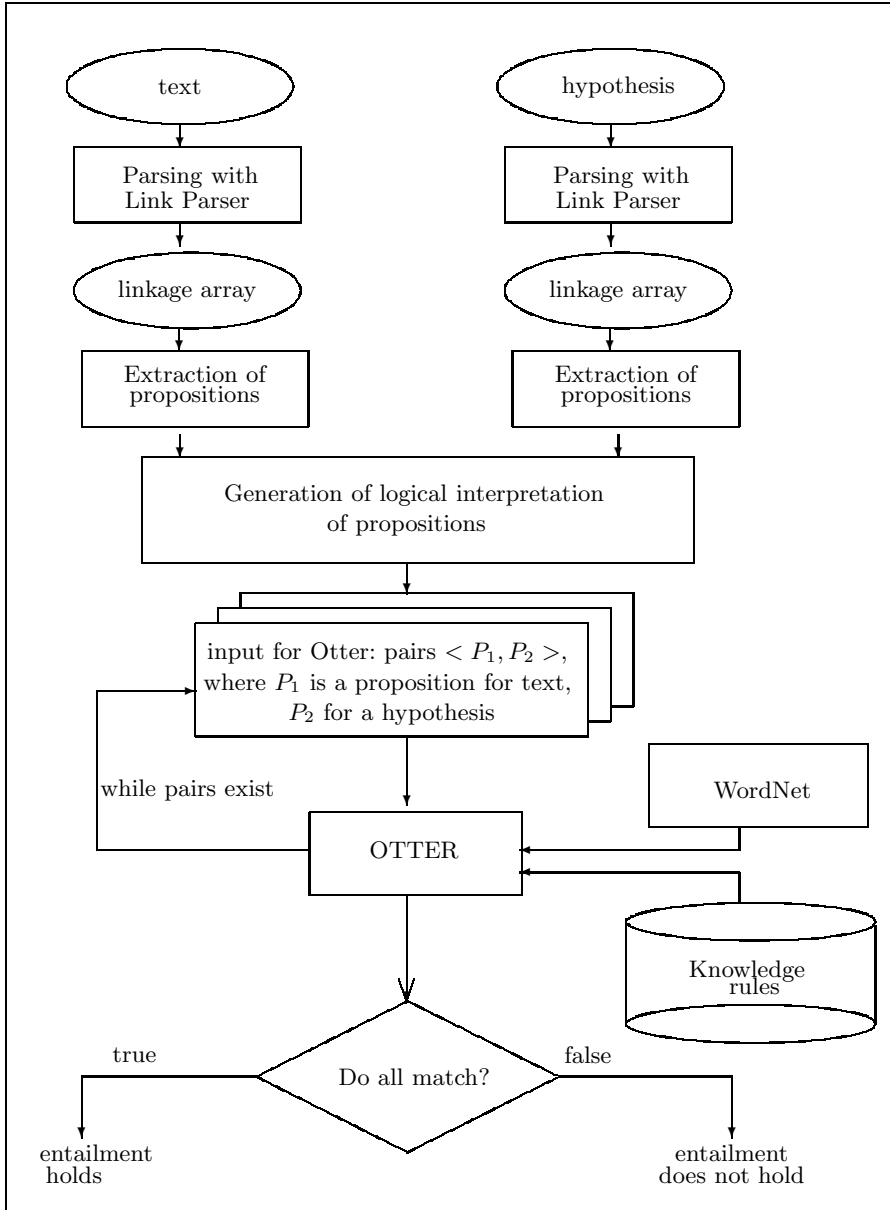


Fig. 1. Architecture of the system

have been extracted the system generates a logical form for every proposition. The logical forms of the propositions are given as an input to the theorem prover Otter. Data returned by Otter allows the system to make a decision about the relation of entailment between the text and the hypothesis.

Figure 2 shows an example the process of making a decision about the entailment relation between the text and the hypothesis. Every proposition of the hypothesis – *coffee gives health benefits* – is compared to all the propositions of the text sentence. In the example, since there is one proposition – *coffee provides health benefits* – that entails the only proposition of the hypothesis, the entailment holds.

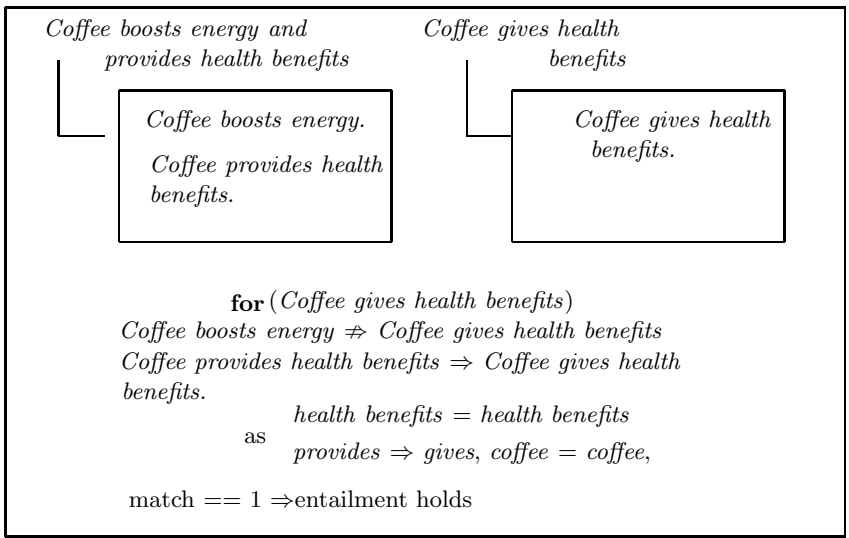


Fig. 2. Comparison of Propositions

5.1 Link Parser and Proposition Extraction

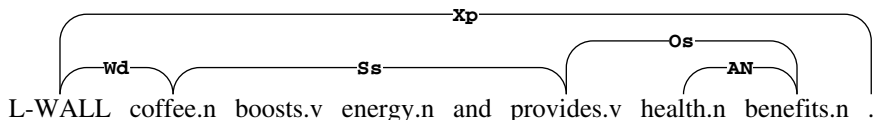
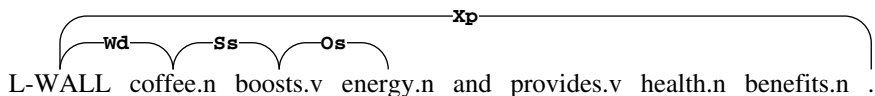
The current version of the system needs a full parse of the text and hypothesis sentences. Both sentences are parsed with the Link Parser. The output of the parser is an array of links (or *linkages*) between words.

In the example,



the subject of the sentence *coffee* is connected to the predicate *boosts* by means of the *Ss* link, that is defined as a relation between a subject in a singular number and its predicate. Capital *S* means subject, small *s* means singular. In a similar way *boosts* and *energy* are connected through the object-predicate linkage *Os*.

The full output of the Link Parser for the text sentence from our example may be represented graphically in the following two linkages:¹



This output of the parser can be easily used to extract the atomic propositions from the sentences. One only needs to define which sets of links constitute a proposition and to check if there are coherent chains of such links in a sentence parse. *Ss* and *Os* links will give us a proposition, links between a noun, participle and its object, like in *boy playing chess*, will also constitute a proposition, and so on. Table 1 shows the basic combinations of links that constitute a proposition.

For our example text sentence two atomic propositions are extracted:

coffee boosts energy and coffee provides health benefits.

The hypothesis sentence contains only one proposition, namely

coffee gives health benefits.

5.2 Logical Forms and Automated Deduction

After the propositions are extracted they are converted into logical forms. These logical forms will constitute an input to the theorem prover. For this reason they are formatted according to the syntactic requirements of the particular theorem prover, Otter in our case.

The current logical representation is flat and syntax-dependent, and could be considered a simplified version of other logical representations proposed in the literature (e.g. Hobbs 1985, Copestake et al. Draft).

There are three types of objects: *Subj(x)*, *Obj(x)*, *Pred(x)*, and a meaning attaching element *iq(x, <meaning of x>)*. With this notation, the proposition *coffee boosts energy* has the following logical representation:

exists *x* exists *y* exists *z* (*Subj(x)* & *iq(x, 'coffee')* & *Pred(y)* & *iq(y, 'boosts')* & *Obj(z)* & *iq(z, 'energy')*).

Also, there are two variants of relationships *attr(x, y)* and *prep(x, y)*; the following lines show an example of their usage:

Somali capital – *Subj(x)* & *iq(x, 'capital')* & *attr(x, y)* & *Subj(y)* & *iq(y, 'somali')*.
 a zoo in Berlin – *Obj(x)* & *iq(x, 'zoo')* & *prep(x, y)* & *Obj(y)* & *iq(y, 'Berlin')*.

¹ Each picture shows one of the branches of the coordination as returned by Link Parser.

Table 1. The list of linkages used to extract propositions (only relevant links shown). For the reference to the meaning of the linkages visit the Link Grammar Documentation web-site <http://bobo.link.cs.cmu.edu/link/dict/index.html>.

Linkages	Example	Proposition
<i>Ss/Sp Os/Op</i>	<p>The boy.n plays.v chess.n</p>	<i>boy plays chess</i>
<i>Mg Os/Op</i>	<p>The boy.n playing.v chess.n is here</p>	<i>boy plays chess</i>
<i>MX (with Xd, Xc)</i>	<p>The boy.n , a chess.n player.n , is here</p>	<i>boy is a chess player</i>
<i>Bs/Bp (with R, RS)</i>	<p>The boy.n who plays.v chess.n is here</p>	<i>boy plays chess</i>
<i>Bs/Bp (with R#, S#)</i>	<p>The boy.n the girl.n likes.v is here</p>	<i>the girl likes the boy</i>
<i>Mv; Mv (with MVp, Js/Jp and by)</i>	<p>The book.n read.v by the boy.n is good</p>	<i>the boy read the book</i>
<i>Ss/Sp Pv MVp Js/Jp with by</i>	<p>The book.n was.v read.v by the boy.n</p>	<i>the boy read the book</i>

As soon as a proposition is converted to its logical representation it is ready to be used as an input for Otter. However, Otter will not be able to prove anything without some background knowledge. The background knowledge provided to Otter comes from three different sources:

1. Knowledge about the relation between concepts, such as *give* and *provide*, or *table* and *furniture*. This is extracted from WordNet.
2. Knowledge about the relationships, like $attr(x, y)$, which could be substituted by $prep(x, y)$ under some circumstances.
3. Knowledge about the different representations of the same concept. A number, for example, could be represented as a word (*one*) or as a numeric expression (*1*).

The word relations provided by WordNet were used to compute a measure of entailment between pairs of words. Concept c_i is in the entailment relation to a concept c_j , if c_i is less generic than c_j (by concept we mean here a particular sense of a word, or synset, using WordNet terminology). In a slightly more formal wording, c_i entails c_j if c_i is equal to c_j or c_i is lower than c_j in the WordNet hyponymy hierarchy. We generalize this idea to allow a degree of entailment between two arbitrary words. This degree of entailment depends on the length of the path between the two words, and the number of senses of each word. The final formula is:

$$rel(c_i, c_j) = p(c_i) \times p(c_j) \times score(c_i, c_j) \tag{1}$$

$$score(c_i, c_j) = C^{length(c_i, c_j) - 1} \tag{2}$$

where $p(c_i)$ is the probability that a word in a sentence is presented by the concept c_i , and $length(c_i, c_j)$ is the length of the path connecting the concepts. The constant C ($0 < C \leq 1$) has to be chosen empirically.

As it could be easily seen from the formula the final score is between zero and one. The idea behind the formula is the following. First, we assume that a word sense disambiguation system gives as an output the probability distribution over all the senses of a word. Secondly, we assert that the relation between concepts becomes weaker as the length of the path connecting the words increases. The constant C helps to adjust the impact of the length between concepts in the final relatedness score.

Thus, for our example we can obtain the rule:

all x ($iq(x, \text{'provide'}) \rightarrow iq(x, \text{'give'})$).

Figure 3 shows the result obtained for the comparison of two lexical concepts, *provide* and *give*.

To account for the sources of information 1. and 2. the system used the following rules:

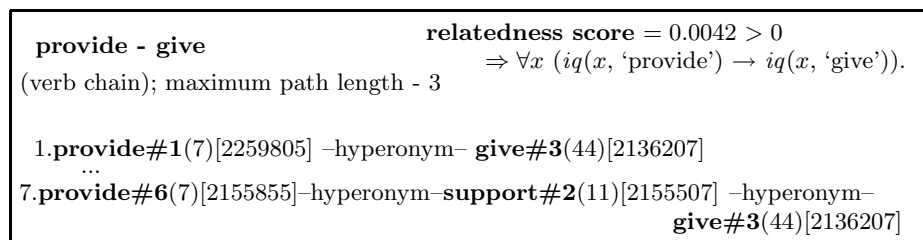


Fig. 3. WordNet relatedness between the concepts *provide* and *give*; data and results. The figure shows the paths connecting the lexical concepts and the final relatedness score.

- a) **Negation:** all x all $x77$ ($nig(x, x77) \rightarrow -iq(x, x77)$).
- b) **Attribute-preposition relation:** all x all y ($attr(x, y) \rightarrow prep(x, y)$).
- c) **Numbers:** all x ($iq(x, 'one') \rightarrow iq(x, '1')$).

Rule b) has been introduced to allow paraphrases of multi-word expressions, such as the ones below, where an attribute of a noun is converted into a prepositional phrase:

night flight \leftrightarrow *flight in (the) night*
pet spray \leftrightarrow *spray for pets*
peanut butter \leftrightarrow *butter from peanut*
abortion problem \leftrightarrow *problem about abortion*

5.3 Proposition Comparison

The text entails the hypothesis if for every proposition in the hypothesis there is one proposition in the text sentence that could entail it, as shown in Figure 2. The decision that proposition p_1 entails proposition p_2 can be easily made after sending Otter both propositions and all the background rules, including those obtained from the WordNet lexical database.

5.4 Results and Performance

The entailments addressed in the program belong to the class of lexico-syntactical entailments discussed in Section 3. As mentioned above, only around 49% of all entailment pairs in the RTE dataset are entailments of this kind. One might expect the system would not jump over the 50% barrier then. It is not true though. The results are different because the system also gets points for recognizing that there is no entailment relation between the text and hypothesis. It almost always states the absence of the entailment relation correctly.

Also the mistakes introduced on the stage of the parsing or during the process of proposition extraction brought some noise into the results, as some of the decisions that were made happened to be correct or incorrect because of the mistakes.

The actual results are shown in Table 2 and summarised here:

cws: 0.5067; accuracy: 0.5188; precision: 0.6119;
 recall: 0.1025; f: 0.175.

The system successfully predicts the following types of entailments:

- **WordNet generalization** T: *The decision is made.* – H: *The determination is made.*
- **Logical generalization** T: *The good decision is made.* – H: *The decision is made.*
- **Syntactic variations** T: *The zoo built recently by the government is open for visitors now.* – H: *The zoo was built.*
- **NN syntactic structure** T: *The Brazilian president visited France.* – H: *The president of Brazil visited France.*

Table 2. Performance of the system according to the *PASCAL Recognizing Textual Entailment Challenge* evaluation method

Task Cws	Accuracy
CD	0.6121 0.5867
IE	0.5519 0.5083
MT	0.4341 0.4917
QA	0.4649 0.4769
RC	0.4702 0.5214
IR	0.5452 0.5200
PP	0.4797 0.5111

Some of the mistakes of the system were caused by the overgeneralization of the logical forms for specific language structures. The two following examples illustrate this:

text. *The gastronomic capital of France is Lyon.*

hypothesis. *The capital of France is Lyon.*

text. *The man came to the park by car.*

hypothesis. *The man came to a car park.*

The first example is a typical intersective reading of adjectives. In general, the *gastronomic capital* of a country is not necessarily the *capital* of the country. However, all the predicates of the logical form of *capital of France* appear in the logical form of *gastronomic capital of France*. Therefore, an entailment is wrongly detected.

In the second example, our approach represents the dependency relation between *car* and *park* differently but the attribute-proposition relation rule makes the logic prover ignore the difference. Again, an entailment is wrongly detected.

6 Comparison with Other Systems

The majority of the systems presented at the PASCAL RTE Challenge tried to approximate entailment by computing lexical relations at the word level. The roughest approximation was found by applying the BLEU metric for the machine translation evaluation (Pérez and Alfonseca 2002) and lexical similarity scores (Jijkoun and de Rijke 2005).

Our system was one of the few approaches that used a logical representation of the input data. An example of another system relying on logic was the one by Bos and Markert (2005). It is interesting to note that ours was the only system where the sentence was divided into minimal semantic elements (“propositions”) so that the subsequent data analysis was applied directly to propositions and not to the original sentences. This division simplifies significantly the process of

creation of a logical representation of the sentence and emphasizes the logical nature of the entailment relation.

We have not proposed here any methods to extract information about entailments in order to construct an entailment knowledge database that may help to the recognition of entailment. One work on the topic deserving attention is the one by MITRE (Burger and Ferro 2005).

7 Summary and Future Work

The present work implements the basic logical property *A and B entails A*. For this, the system builds the logical form of each independent atomic proposition from the text and hypothesis sentences, and sends the logical forms to the Otter automatic prover. The information sent to Otter is extended with general axioms and background information about the words used by the text and hypothesis. WordNet is used as a lexical resource to detect the degree of entailment between individual words. Overall, the system is simple, based on fundamental concepts, and can be seen as a baseline on which to try to recognize more complex types of textual entailment.

This paper also presented a discussion of the nature of textual entailment with respect to the various sources of entailment and the types of entailments depending on the tools required for their recognition.

The next step is to work on a more accurate implementation of the algorithm, namely to include the analysis of more complex verb and noun groups, in particular with respect to the handling of prepositional attachments. We also plan to work on the background information that needs to be sent to the theorem prover. Finally, we are investigating alternative ways of assessing semantic similarity on the word level and its relation to entailment and other logical concepts.

We believe that our work shows the feasibility of the combined use of logical forms with lexical resources to detect the entailment between two sentences.

References

- Bos, J., Markert, K.: Combining Shallow and Deep NLP Methods for Recognizing Textual Entailment. Proceedings of the PASCAL Recognising Textual Entailment Challenge (2005) 65–68
- Budanitsky, A., Hirst, G.: Semantic Distance in WordNet: An Experimental, application-oriented evaluation of five measures. Proceedings of the NAACL–2001 Workshop on WordNet and Other Lexical Resources. Pittsburg PA. (2001)
- Burger, J., Ferro, L.: Generating an Entailment Corpus from News Headlines. Proceedings of the Empirical Modelling of Semantic Equivalence and Entailment, ACL-2005 (2005) 49–54
- Copestake, A., Flickinger, D. Sag, I.A. Minimal Recursion Semantics: an Introduction (on-line draft). <http://lingo.stanford.edu/sag/publications.html>
- Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. Proceedings of the PASCAL Recognising Textual Entailment Challenge (2005)

- Hobbs, J.R.: Ontological Promiscuity. *Proceedings of the Association for Computational Linguistics* (1985) 61–69
- Jijkoun, V., de Rijke, M.: Recognizing Textual Entailment Using Lexical Similarity. In the *Proceedings of the PASCAL Recognising Textual Entailment Challenge* (2005) 73–76
- Jurafsky, D., Martin, J.H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice-Hall. Chapters 14–15 (2000)
- Kalman, J.A.: *Automated Reasoning with OTTER*. Rinton Press. Paramus, NJ (2001)
- Mahesh, K., Nirenburg, S., Cowie, J., Farwell, D. An Assessment of Cyc for Natural Language Processing. CRL technical report MCCS-96-302, New Mexico State University (1996)
- Miller, G.: WordNet: A lexical database for English. *Communications of the ACM*, 38(11) (1995) 39–41
- Mollá, D., Hutchinson, B. Intrinsic versus Extrinsic Evaluations of Parsing Systems. *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL), workshop on Evaluation Initiatives in Natural Language Processing* (2003) 43–50.
- Pérez, D., Alfonseca, E.: 2005. Application of BLEU algorithm for Recognizing Textual Entailments. In the *Proceedings of the PASCAL Recognising Textual Entailment Challenge* (2002) 9–12
- Quirk, R., Greenbaum, S., Leech, G., Svartvik, J. *A Comprehensive Grammar of the English Language*. Longman, London (1985)
- Sleator, D., Temperley, D.: *Parsing English with a Link Grammar*. Carnegie Mellon University Computer Science technical report CMU-CS-91-196 (1991)
- Sutcliffe, R.F.E., Koch, H., McElligott, A., editors. *Industrial Parsing of Software Manuals*. Rodopi, Amsterdam (1996).
- Vanderwende, L., Coughlin, D., Dolan, B.: What Syntax can Contribute in Entailment Task. In the *Proceedings of the PASCAL Recognising Textual Entailment Challenge*, (2005) 13–16