Dependency-based Textual Entailment

Vasile Rus
Department of Computer Science
Institute for Intelligent Systems
The University of Memphis
Memphis, TN 38120, USA
vrus@memphis.edu

Abstract

This paper studies the role of dependency information for the task of textual entailment. Both the Text and Hypothesis of an entailment pair are mapped into sets of dependencies and a score is computed that measures the similarity of the two sets. Based on the score an entailment decision is made. Two experiments are conducted to measure the impact of dependencies on the entailment task. In one experiment we compare the dependency-based approach with a baseline approach on a standard data set. In a second experiment, we measure the performance on a subset of the standard data set. The subset is so selected to minimize the effect of other factors, such as word-level information, in the performance measurement process. A brief discussion compares the dependency-based approach to other, similar approaches.

Introduction

The task of textual entailment is to decide whether a text fragment the size of a sentence, called the Text (T), can logically infer another text of same or smaller size, called the Hypothesis (H).

Entailment has received a lot of attention since it was proposed under the Recognizing Textual Entailment (RTE) Challenge (Dagan, Glickman, & Magnini 2004 2005) in 2004. In our experiments presented here, we use the standard data set that RTE offers for development and comparison purposes. Below, we illustrate the Text and Hypothesis of pair 2028 from the RTE test.

Text: *Besancon is the capital of France's watch and clock-making industry and of high precision engineering.*
Hypothesis: *Besancon is the capital of France.*

For this particular pair the entailment decision is FALSE since H cannot be logically inferred from T.

Textual Entailment is a hard task that requires linguistic, world and domain knowledge to be solved. In this paper, we study the impact of dependency relations on the task of textual entailment. Both the Text and Hypothesis are mapped into two sets of dependencies: the T-set and H-set, respectively. A score is then computed that measures the degree to which the dependencies in the H-set are present in the T-set. Based on the score, an entailment decision is made. Dependencies are labelled relations among words in a sentence. For example, there is a *subj* dependency between *capital* and *Besancon* in the previous example which we represent as the triplet *(subj, capital, Besancon)*. The first term in the triplet is the name or label of the relation, the second term represents the head word, and the last term is the dependent word. We use MINIPAR (Lin 1998), an automated dependency parser, as our primary source of dependency information. Its output is further processed to obtain dependencies triplets, and to filter out irrelevant information. Details about how to map the Hypothesis and Text into sets of dependencies are provided in subsequent sections.

The major advantage of using dependencies over word-based similarity measures, such as the one described in (Monz & de Rijke 2001), is that they capture syntactic information which is important in fine language understanding tasks, such as textual entailment. Syntactic relations are important to decide that *Yahoo took over Overture* entails *Yahoo bought Overture* (example from first RTE Challenge task description) and does not entail *Overture bought Yahoo*. Using dependencies the latter is rejected simply because the subject of *buying* is *Overture* which contradicts the Text where *Overture* is the object of *took over* while the subject is *Yahoo*.

Another advantage of using a dependency-parser instead of a phrase parser (which returns phrases hierarchically organized in a parse tree) is its applicability to a larger variety of languages. It is known that a phrase-based parser is not applicable to free-order languages. For our dependency-based representation of syntactic information all it takes to port the system to a new language is to train the dependency extractor for that particular language and plug-it in our system, by simply replacing MINIPAR.

To preview our results we show that dependencies lead to good results for data where solely lexical/word information is not enough.

The rest of the paper is structured as follows. The next section presents related work on the entailment task. Section *The Dependency-based Approach* describes our approach to solving the entailment task. Section *Experiments and Results* outlines the two experiments we conducted and the results we obtained. A comparative discussion is also included in this section. The paper ends with *Conclusions*.

Related Work

A great number of approaches to entailment have been taken since the RTE (Dagan, Glickman, & Magnini 2004 2005) data was made available, most of them after 2004. They range from shallow approaches such as weighted-bag of words to deeper approaches that rely on theorem proving and world knowledge.

It is not the purpose of this paper to carefully review these approaches. Rather, we briefly present attempts prior to the RTE Challenge and discuss work that presents detailed analysis of RTE-like entailment, noting that it is difficult to get a comprehensive analysis of particular aspects of recently analyzed entailment.

In one of the earliest explicit treatments of entailment Monz and de Rijke (Monz & de Rijke 2001) proposed a weighted bag of words approach to entailment. They argued that traditional inference systems based on first order logic are restricted to yes/no decisions when it comes to entailment tasks while their approach delivered 'graded outcomes'. They established entailment relations among larger pieces of text - on average segments of 4 sentences - than the proposed RTE setup where the text size is a sentence (seldom two) or part of a sentence (phrase).

A closely related effort is presented in (Moldovan & Rus 2001). They show how to use unification and matching to address the answer correctness problem. Answer correctness can be viewed as entailment: Is a candidate answer entailing the ideal answer to the question? Initially, the question is paired with an answer from a list of candidate answers (obtained through some keyword proximity and shallow semantics methods). The resulted pair is mapped into a first-order logic representation and a unification process between the question and the answer follows. As a back-off step, for the case when no full unification is possible, the answer with highest unification score is top ranked. The task they describe is different than the RTE task because a list of candidate answers to rank are available. The granularity of candidate answers and questions is similar to the RTE data.

Recently, Dagan and Glickman (Dagan & Glickman 2004) presented a probabilistic approach to textual entailment based on lexico-syntactic structures. They use a knowledge base with entailment patterns and a set of inference rules. The patterns are composed of a pattern structure (entailing template → entailed template) and a quantity that tells the probability that a text which entails the entailing template also entails the entailed template.

Pazienza and colleagues (Pazienza, Pennacchiotti, & Zanzotto 2005) use syntactic graph distance approach for the task of textual entailment. Their approach is closest to ours. By comparison, we use a different scoring mechanism and a different set of syntactic relations. Our score is simple and thus easy to compute and easy to interpret. The set of syntactic relations we use is a reduced set of the relations handled by MINIPAR.

Vanderwende and colleagues (Vanderwende, Coughlin, & Dolan 2005) looked at T-H pairs from the RTE data that could be solved using "solely" syntactic information. Their broad definition of the meaning of syntax spans over things traditional considered in other linguistic subareas (pronoun resolution is usually part of discourse and not syntax). For example, it includes argument structure, pronoun resolution and alternations. They claim that a 'large proportion of the data', i.e. 37% of the test pairs, can be handled with syntax alone and that adding a general-purpose thesaurus can boost the performance to 49%. The claims are based on two human annotators who examined the data manually.

Bar-Haim and colleagues (Bar-Haim, Szpektor, & Glickman 2005) present a very interesting conceptual analysis of entailment at lexical and syntactic level. They adopt same broad definition of syntax as Vanderwende and colleagues. They found that paraphrases are important and that a lexico-syntactic model outperforms a lexical model. Both the lexico-syntactic and lexical models have low recall. Their work too, is based on manual processing of the test data.

We present here a fully automated system that implements a dependency-based approach to entailment. We answer two questions: what is the performance of a dependency-based approach and what is the impact of dependencies on the performance of the system on top of word-similarity approaches.

The Dependency-based Approach

Before we proceed let us remind the reader that the entailment task as defined by RTE is a binary classification task in which the output can have two values: TRUE, meaning the Text entails the Hypothesis, or FALSE, otherwise.

Our approach starts by mapping both the Text and the Hypothesis into sets of dependencies. It continues with a step in which it computes how many dependencies in the Hypothesis are present in the Text. The result is normalized by the total number of dependencies in the Hypothesis leading to an *entailment score*. If all H-dependencies are present, it means the Hypothesis is syntactically (and lexically since the related words in a dependency need to match also) contained by the Text and thus we can conclude the Text entails the Hypothesis. Otherwise, the entailment score is further analyzed to draw the best possible decision. If the score is above 50% we decide TRUE, if less we decide FALSE[1]. Besides

[1]Percentages and their equivalent values between 0 and 1

the simple TRUE/FALSE decision we assign a degree of confidence in the decision. The entailment score is also used to obtain the confidence measure. We use three levels of confidence obtained as described in the following: entailment scores of 90% or higher or 10% or lower lead to 100% confidence; scores between 90% and 75% or between 10% and 25% lead to 75% confidence. Everything else leads to 50% confidence. Here are two illustrative cases. For a score of 91%, meaning 91% of the Hypothesis dependencies are found among the Text dependencies, then we conclude that the Text entails the Hypothesis (TRUE). Our confidence in this case would be 100% since the score is above 90%. In a second case, if only 7% of the Hypothesis dependencies are found among the Text dependencies we conclude the Text does not entail the Hypothesis (FALSE). Since the dependency-based score is below 10% our confidence in the FALSE entailment decision is strong, i.e. 100%. The thresholds were empirically learned.

Let us look closer at how an entailment pair is mapped into sets of dependencies and how we measure the degree of containment. The mapping comprises of four steps: (1) preprocessing, (2) part of speech tagging, (3) dependency parsing, and (4) postprocessing. The preprocessing step involves tokenization and lemmatization. Tokenization is the process of separating punctuation from words. Lemmatization maps morphological variations of a word to a canonical form. Stemming is another process of reducing morphological variations of words to same base form. As opposed to lemmatization, stemming sometimes results in a non-English word. This could be a problem if, for example, we want to use synonyms (words with same meaning) for better dependencies mapping. The second step, part of speech tagging, assigns parts of speech to each word in a sentence. The dependency parsing step maps a sentence into a set of dependencies. The postprocessing step further refines the dependencies obtained from the dependency parser and is detailed in section *Dependency Parsing and Refinement*.

Once the sets of dependencies, one for T and one for H, are obtained we scan each relation in the H-set and find its corespondent in the T-set. A relation has a match if the name of the relation and the related words are identical. The position of the words is also important: the second term in the H dependency triplet must match the second term in the T dependency triplet, etc.

Dependency Parsing and Refinement

In this section we detail how to map English sentences into dependency triplets based on MINIPAR and a refinement procedure that we developed.

The mapping comprises two steps:

- parsing with MINIPAR and lemmatize;
- post-processing the output of MINIPAR;

are both used throughout the paper to represent the score. For instance a score of 50% is equivalent to 0.50, a score of 7% is equivalent to 0.07, etc.

We run MINIPAR on the RTE sentences, both Text and Hypothesis. We use the lemmatizing and relation output options. This insures that the output is formatted as pairs of words with the syntactic relation between them, and these words are lemmatized. We show below the output obtained for the hypothesis sentence: *Besancon is the capital of France.*

```
fin   C:i:VBE be
be      VBE:s:N Besancon
be      VBE:pred:N      capital
capital N:subj:N        Besancon
capital N:det:Det       the
capital N:mod:Prep      of
of      Prep:pcomp-n:N  France
```

It is interesting to notice that when the main verb of the sentence is *be*, MINIPAR will consider the predicate to consist of *be* and the verb complement, and it will connect the subject with the complement, bypassing the verb. This is a good feature for textual entailment, as it will help address situations such as the one when a noun modifier in the Text becomes a complement of the verb *be* in the Hypothesis:

Text: *The Alameda Central, west of the Zocalo, was created in 1592.*
Hypothesis: *The Alameda Central is west of the Zocalo.*

The parse above also shows why we need a post-processing step. First of all, we filter out pairs such as: "fin C:i:VBE be", since it is not informative as far as dependency pairs are concerned. We also filter out determiner-noun and auxiliary-verb pairs. Second, we compress two or more pairs, such that we obtain only pairs containing open-class words (closed-class words are mainly prepositions and conjunctions, the other are open-class). For example, we combine

```
capital N:mod:Prep      of
of      Prep:pcomp-n:N  France
```

to produce the dependency *(of,capital,France)*. This type of compression is performed for pairs containing prepositions and clause subordinators and coordinators.

The Scoring

The formula to obtain an overall score aims to deliver both a numerical value for the degree of entailment between T and H and a degree of confidence in our decision. The scores range from 0 to 1, with 1 meaning TRUE entailment with maximum confidence and 0 meaning FALSE entailment with maximum confidence. The score is so defined to be non-reflexive, i.e. $entail(T,H) \neq entail(H,T)$.

The formula to compute the score is provided by Equation 1. D_h represents a single dependency in H_d, the set of dependencies in the Hypothesis, and D_t represents a single dependency in T_d, the set of dependencies in the Text. From the way the score is defined it is obvious that $entscore(H,T) \neq entscore(T,H)$. The match

$$entscore(T,H) = \frac{\sum_{D_h \in H_d} max_{D_t \in T_d} match(D_h, D_t)}{|H_d|} \quad (1)$$

function returns 1 if its arguments are identical (or synonyms) and 0 otherwise. For a single H-dependency we return the max of all matching results between itself and all T-dependencies. We did not experiment with partial matching functions but that would be an interesting aspect to try. We could, for instance, return a .66 percent matching score for two dependencies that share a word (in same position) and the relation label (one word being mismatched).

Experiments and Results

Let us describe first the experimental setup as defined by RTE. Then, we show the two experiments we conducted to evaluate the performance of the dependency-based approach.

The dataset of text-hypothesis pairs was collected by human annotators. It consists of seven subsets, which correspond to typical success and failure settings in different applications: Question Answering (QA), Information Retrieval (IR), Comparable Documents (CD), Reading Comprehension (RC), Paraphrase Aquisition (PP), Information Extraction (IE), Machine Translation (MT). Within each application setting the annotators selected both positive entailment examples (judged as TRUE), where T does entail H, as well as negative examples (FALSE), where entailment does not hold (roughly 50%-50% split).

The evaluation is automatic. The judgements (classifications) returned by the system are compared to those manually assigned by the human annotators (the gold standard). The percentage of matching judgements provides the *accuracy* of the run, i.e. the fraction of correct responses. As a second measure, a *Confidence-Weighted Score* (*CWS*, also known as average precision) is computed. Judgements of the test examples are sorted by their confidence (in decreasing order from the most certain to the least certain), calculating the following measure:

$$\frac{1}{n} * \sum_{i=1}^{n} \frac{\#-correct-up-to-pair-i}{i} \quad (2)$$

where n is the number of the pairs in the test set, and i ranges over the pairs. The Confidence-Weighted Score varies from 0 (no correct judgements at all) to 1 (perfect score), and rewards the systems' ability to assign a higher confidence score to the correct judgements than to the wrong ones. A third reported measure is *precision* which is the accuracy for only TRUE-solved pairs (true positives).

Experiment 1

In this experiment we aim to find out what is the performance of our dependency-based approach as compared to a baseline approach. Since the test data is balanced, an approach that consistently guesses FALSE or TRUE leads to 50% accuracy. The first two lines in Table 1 represents the results for guessing FALSE or TRUE, respectively. The difference between the results in the two rows is in the Precision column which is 0 for guessing FALSE and 1 for guessing TRUE. The third line represents the results of the dependency-based approach presented here for the entire test data set. The rest of the table depicts the results by application type. The precision is not reported for individual applications because the scoring software provided by RTE does not report it.

Approach/Applic.	CWS	Accuracy	Precision
baseline-F	0.4906	.5000	0.000
baseline-T	0.5094	.5000	1.000
dep	0.5350	0.5270	0.5311
CD	0.7092	0.6600	
IE	0.5985	0.5583	
MT	0.4721	0.4583	
QA	0.3477	0.4000	
RC	0.4400	0.4500	
PP	0.4402	0.4800	
IR	0.5419	0.5333	

Table 1: Results from Experiment 1 on the entire RTE test data set. The bottom part depicts results by individual application.

The results on this first experiment are only slightly better than the baseline of consistently guessing TRUE or FALSE. As we show in the *Discussion* section those figures are similar to other systems. The results can be explained by errors that occur in each processing step: preprocessing, part-of-speech tagging, MINIPAR, and postprocessing. On top of that, many entailment pairs cannot be solved simply using linguistic information. World and domain knowledge is needed which we do not use at all in our approach. The major research question we ask is how good a solely-dependency based approach can be. Exploring how to integrate word knowledge on top of a purely linguistic approach is a challenging task which we plan to tackle in the future.

Experiment 2

In this second experiment we estimate the impact of dependency information on a subset of RTE test data that we believe better illustrates the contribution of dependencies on top of lexical matching.

We isolated a subset of the test data that contains pairs with perfect word (lexical) overlap - that is, all

the words in H had a correspondent in T. We then manually checked whether syntactic information could help solve the entailment for this particular T-H pairing. The isolation yielded 106 pairs with perfect word overlap of which 101 were selected for further analysis. Five pairs were discarded as the annotator disagreed with the RTE answer. Finally, we measure the performance of our system on the subset that was manually determined to benefit from syntax. Because lexical information cannot help in those cases it is a good testbed to check how much dependencies help.

We judged the lexically overlapped pairs to fall into one of three syntactical categories. These categories require some explanation which we have supplemented with condensed examples taken from the RTE.

S1-type pairs were deemed those where relatively simple syntactical rules were judged likely to bring a significant number of correctly identified entailments. For example, consider the following text and hypothesis.

Text: *The Alameda Central, west of the Zocalo, was created in 1592.*
Hypothesis: *The Alameda Central is west of the Zocalo.*

Replacing the comma that separates two consecutive NPs in the text-half of a pair with the verb be, is likely to result in entailment. The rule is concise, simple to apply, and predicted to be generally correct. We therefore labeled examples of this kind as type-S1.

S2-type pairs differed from S1 pairs in two ways. First, S2-type pairs were deemed those where describing the syntax rule was moderately complex. Second, and assuming the complexity of the rule did not prevent it from being computationally realized, a S2-type pair was deemed one where the likelihood of the rule identifying correct entailments was significantly lower than that of S1-types. For example, consider the following text and hypothesis.

Text: *In 1541, the Turks took Buda and held it until 1686; the city changed very little during this time.*
Hypothesis: *The Turks held Buda between 1541 and 1686.*

In this example, the prepositions *in* and *until* in the text-half correspond to the prepositions *between* and *and* in the hypothesis. However, the overlap of these prepositions is not enough by itself. Issues such as the presence and position of each token, and corresponding dates must also be considered. There is also the complicating matter of the presence of a coordinating conjunction in the text-half, meaning that the rule is complex and unlikely to bring as significant a number of correctly identified entailments as the S1-type.

S3-type pairs fell into two categories. First, pairs were deemed S3 if the hypothesis required extra textual information. For example, consider the following pair.

Text: *A state of emergency was declared in Guatemala City.*
Hypothesis: *A state of emergency was declared in Guatemala.*

Although the entailment is correct in this hypothesis, there is no syntactical way to know that Guatemala City is in Guatemala.

Pairs were also deemed S3 if a potential syntactical rule was deemed highly complex and unlikely, even if constituted, to bring a significant number of correctly identified entailments. For example, consider the following pair.

Text: *The demonstrators were calling for a trial of the right-wing politician and on seeing them he gave them a rude gesture and the police then had to stop four demonstrators who tried to chase the senator.*
Hypothesis: *Demonstrators gave the right-wing senator a rude gesture.*

This hypothesis calls for both pronoun resolution and syntactical agreement over distant sentence elements. Even if a rule could be formulated, its likelihood of correctly identifying a significant number of entailments was deemed small.

Approach/Applic.	CWS	Accuracy	Precision
baseline-F	0.5465	.5957	0.000
baseline-T	0.4535	.4043	1.000
dep	0.6074	0.6170	0.5556
CD	0.7133	0.4000	
IE	0.6381	0.7000	
MT	1.0000	1.0000	
QA	0.6268	0.7500	
RC	0.4103	0.5294	
PP	0.0000	0.0000	
IR	1.0000	1.0000	

Table 2: Results from Experiment 2 on a subset of the RTE test data set. The bottom part depicts results by individual application.

Let us look at the figures. Of the original 106 pairs, 44 solved to TRUE and 62 to FALSE according to RTE. Of the final 101 pairs, 41 led to TRUE and 60 to FALSE meaning a blind method that guesses FALSE for this category could deliver .5940 accuracy. This is in concordance with (Dagan, Glickman, & Magnini 2004 2005) which reports that pairs with high lexical match are biased towards FALSE entailment. About 47 (47%) of the 101 could be answered by syntax as defined by us (S1) and 69 (72%) by a broader definition of syntax (S1+S2). This is a little bit more optimistic than results reported by Vanderwende and colleagues (Vanderwende, Coughlin, & Dolan 2005) but close to that.

Following the manual check, we evaluated the performance of our system on the 47 pairs that could be answered by syntax. These pairs were split as: 19-TRUE and 28-FALSE according to RTE annotation. Since those pairs have perfect word overlap a lexical

approach similar to (Monz & de Rijke 2001), which relies only on word-level information and no syntactic information, is useless/blind to this subset. The relation information that is found in dependencies could help and we evaluated to what extent.

Table 2 summarizes the results obtained on this data subset. The difference between the two baselines in rows 1 and 2 is significant since the data subset is skewed, i.e. it has more FALSE cases. An average of the two would probably be a better baseline in this experiment. The cws and accuracy measures for the dependency-based approach, shown in the third row, are above .60 which is significantly better than the results in Experiment-1 and better than an average of the two baselines in Table 2. Precision is high (.55) as compared to the first Experiment although this particular data subset is skewed: has fewer TRUE solved pairs. Actually, this is a good indication that dependencies are helpful, especially at recognizing true positives (TRUE entailment cases). For some particular applications, for instance IR, both measures cws and accuracy are 1 which is misleading. It only means that just a few cases for those particular tasks ended in the small data subset used in this second experiment.

Discussion

system	cws	accuracy
Zanzotto (Rome-Milan)	0.557	0.524
Punyakanok	0.569	0.561
Andreevskaia	0.519	0.515
Jijkoun	0.553	0.536

Table 3: Performance and comparison of different approaches on RTE test data.

In this section we compare our approach with approaches that use similar resources. The reader should keep in mind that all we use is lemmatization and dependencies among pair of words in a sentence. We also use minimal lexical semantics (only synonymy) and no deeper representations of meaning, no reasoning and no world knowledge. The systems shown in Table 3 were selected from the participating systems in the RTE Challenge (Dagan, Glickman, & Magnini 2004 2005) based on the type of resources they use: word overlap, WordNet and syntactic matching. Some of the systems in Table 3 use slightly more resources than we do (they use WordNet more extensively than we do). When compared to the shown systems, our approach has comparable performance on the overall test data set. The results from the second Experiment are significantly better than an average of the two baselines which demonstrates the utility of dependencies on top of word-level information. A direct comparison of our results in the second Experiment, which are clearly better, with the systems in Table 3 is not possible because the figures in the table are not on the data subset that we selected.

Conclusions

This paper studied the role of dependencies for the task of textual entailment. We conducted two experiments on a standard data set and concluded the dependency information leads to good results for data with high word overlap for which solely word information is helpless.

ACKNOWLEDGEMENTS

The authors would like to thank Philip McCarthy for annotating and analyzing the data subset used in the second experiment. We are also thankful to Viviana Nastase for running the MINIPAR parser and the post-processing software for us. Without their help and patience this work would have not been possible.

References

Bar-Haim, R.; Szpektor, I.; and Glickman, O. 2005. Definition and analysis of intermediate entailment levels. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, 43rd Annual Meeting of The Association for Computational Linguistics*. Ann Arbor, MI: 43rd Annual Meeting of The Association for Computational Linguistics.

Dagan, I., and Glickman, O. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. In *Proceedings of Learning Methods for Text Understanding and Mining*.

Dagan, I.; Glickman, O.; and Magnini, B. 2004 - 2005. Recognizing textual entailment. In *http://www.pascal-network.org/Challenges/RTE*.

Lin, D. 1998. Dependency-based evaluation of minipar.

Miller, G. 1995. WordNet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Moldovan, D., and Rus, V. 2001. Logic form transformation of wordnet and its applicability to question answering. In *Proceedings of the ACL Conference (ACL-2001)*.

Monz, C., and de Rijke, M. 2001. *Light-Weight Entailment Checking for Computational Semantics*. 59–72.

Pazienza, M.; Pennacchiotti, M.; and Zanzotto, F. 2005. Textual entailment as syntactic graph distance: A rule based and svm based approach. In *Proceedings of the Recognizing Textual Entaiment Challenge Workshop*.

Vanderwende, L.; Coughlin, D.; and Dolan, B. 2005. What syntax can contribute in entailment task. In *Proceedings of the Recognizing Textual Entaiment Challenge Workshop*.