A Syntactic Textual Entailment System Based on Dependency Parser

Partha Pakray[1], Alexander Gelbukh[2], and Sivaji Bandyopadhyay[1]

[1] Computer Science and Engineering Department,
Jadavpur University, Kolkata, India
parthapakray@gmail.com, sbandyopadhyay@cse.jdvu.ac.in
[2] Center for Computing Research, National Polytechnic Institute,
Mexico City, Mexico
gelbukh@gelbukh.com

Abstract. The development of a syntactic textual entailment system that compares the dependency relations in both the text and the hypothesis has been reported. The Stanford Dependency Parser has been run on the 2-way RTE-3 development set and the dependency relations obtained for a text and hypothesis pair has been compared. Some of the important comparisons are: subject-subject comparison, subject-verb comparison, object-verb comparison and cross subject-verb comparison. Corresponding verbs are further compared using the WordNet. Each of the matches is assigned some weight learnt from the development corpus. A threshold has been set on the fraction of matching hypothesis relations based on the development set. The threshold score has been applied on the RTE-4 gold standard test set using the same methods of dependency parsing followed by comparisons. Evaluation scores obtained on the test set show 54.75% precision and 53% recall for YES decisions and 54.45% precision and 56.2% recall for NO decisions.

Keywords: Textual Entailment, Dependency parsing, Dependency Relations, RTE-3 development set, RTE-4 gold standard test set.

1 Introduction

Recognizing Textual Entailments (RTE) is one of the recent challenges of Natural Language Processing (NLP). Textual Entailment is defined as a directional relationship between pairs of text expressions, denoted by T – the entailing "Text", and H- the entailed "Hypothesis". T entails H if the meaning of H can be inferred from the meaning of T, as would typically be interpreted by people. For instance, the following is a correct entailment pair:

T: US Secretary of State Condoleezza Rice has been defending President Bush's Iraq strategy at a Senate hearing.
H: Rice defends Bush.

There were three Recognizing Textual Entailment competitions RTE-1 in 2005, RTE-2 in 2006 and RTE-3 in 2007 which were organized by PASCAL (Pattern Analysis, Statistical Modeling and Computational Learning) - the European Commission's

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 269–278, 2010.

IST-funded Network of Excellence for Multimodal Interfaces. In 2008, the fourth edition (RTE-4) of the challenge was organized by NIST (National Institute of Standards and Technology) in Text Analysis Conference (TAC). In every new competition several new features of RTE were introduced. The RTE-5 challenge in 2009 includes a separate search pilot along with the main task.

The first PASCAL Recognizing Textual Entailment Challenge (RTE-1) [1], introduced the first benchmark for the entailment recognition task. The RTE-1 dataset consists of manually collected text fragment pairs, termed text (t) (1-2 sentences) and hypothesis (h) (one sentence). The systems were required to judge for each pair whether t entails h. The pairs represented success and failure settings of inferences in various application types (termed "tasks").

In RTE-1 the various techniques used by the participating systems were word overlap, WordNet, statistical lexical relation, world knowledge, syntactic matching and logical inference.

After the success of RTE-1, the main goal of the RTE-2, held in 2006 [2], was to support the continuity of research on textual entailment. The RTE-2 data set was created with the main focus of providing more "realistic" text-hypothesis pair. As in the RTE-1, the main task was to judge whether a hypothesis H is entailed by a text T. The texts in the datasets were of 1-2 sentences, while the hypotheses were one sentence long. Again, the examples were drawn to represent different levels of entailment reasoning, such as lexical, syntactic, morphological and logical.

The main task in the RTE-2 challenge was classification – entailment judgment for each pair in the test set that represented either entailment or no entailment. The evaluation criterion for this task was accuracy – the percentage of pairs correctly judged. A secondary task was created to rank the pairs based on their entailment confidence. A perfect ranking would place all the positive pairs (for which the entailment holds) before all the negative pairs. This task was evaluated using the average precision measure [3], which is a common evaluation measure for ranking in information retrieval.

In RTE-2 the techniques used by the various participating systems are Lexical Relation/ database, n-gram/ subsequence overlap, syntactic matching/ Alignment, Semantic Role labelling/ Framenet/ PropBank, Logical Inference, Corpus/web-based statistics, machine learning (ML) Classification, Paraphase and Templates, Background Knowledge and acquisition of entailment corpus.

The RTE-3 data set consisted of 1600 text-hypothesis pairs, equally divided into a development set and a test set. The same four applications from RTE-2 – namely IE, IR, QA and SUM – were considered as settings or contexts for the pair's generation. 200 pairs were selected for each application in each data set. Each pair was annotated with its related task (IE/IR/QA/SUM) and entailment judgment (YES/NO).

In addition, an optional pilot task, called "Extending the Evaluation of Inferences from Texts" was set up by the NIST, in order to explore two other sub-tasks closely related to textual entailment: differentiating unknown entailment from identified contradictions and providing justifications for system decisions. In the first sub-task, the idea was to drive systems to make more precise informational distinctions, taking a three-way decision between "YES", "NO" and "UNKNOWN", so that a hypothesis being unknown on the basis of a text would be distinguished from a hypothesis being shown false/contradicted by a text.

In RTE-4, no development set was provided, as the pairs proposed were very similar to the ones contained in RTE-3 development and test sets, which could therefore be used to train the systems. Four applications – namely IE, IR, QA and SUM – were considered as settings or contexts for the pair generation. The length of the H's was the same as in the past data sets (RTE-3); however, the T's were generally longer. A major difference with respect to RTE-3 was that the RTE-4 data set consisted of 1000 T-H pairs, instead of 800.

In RTE-4, the challenges were classified as two-way task and three-way task. The two-way RTE task was to decide whether:

- T entails H - in which case the pair will be marked as ENTAILMENT;
- T does not entail H - in which case the pair will be marked as NO ENTAILMENT.

The three-way RTE task was to decide whether:

- T entails H - in which case the pair was marked as ENTAILMENT
- T contradicts H - in which case the pair was marked as CONTRADICTION
- The truth of H could not be determined on the basis of T - in which case the pair was marked as UNKNOWN

In RTE-4 competition [4], 45 runs were submitted by 26 participants, half of whom chose the 3-way task. In the 3-way task, the best accuracy was 0.685. The 3-way task appeared to be altogether quite challenging, as the average 3-way score was 0.51, quite low compared to the results achieved in previous campaigns. The systems performed better in the 2-way task, achieving accuracy scores which ranged between 0.459 and 0.746. These results are lower than those achieved in RTE-3 challenge, where the accuracy scores ranged from 0.49 to 0.80, even though a comparison is not really possible as the data sets were actually different.

In the present paper, a 2-way syntactic textual entailment recognition system has been described that has been trained on the 2-way RTE-3 development set and then tested on the RTE-4 test set. Related works are described in Section 2. Section 3 describes syntactic based RTE system architecture. The experiment carried out on the development and test data sets are described in Section 4 along with the results. The conclusions are drawn in Section 5.

2 Related Works

In the various RTE Challenge, several methods are applied on the textual entailment task. Most of these systems use some sort of lexical matching (e.g. n-gram, word similarity), be it simple word overlap. A number of systems represent the texts as parse trees (e.g. syntactic, dependency) before the actual task. Some of the systems use semantic relation (e.g. logical inference, Semantic Role Labeling) for solving the text and hypothesis entailment problem.

The work presented in [5] suggests that sentence structure plays an important role in recognizing textual entailment and paraphrasing accurately. The Recognizing Textual Entailment System in [6] was based on the use of a broad-coverage parser to extract dependency relations and a module which obtains lexical entailment relations

from WordNet. The use of syntactic tree editing distance to detect entailment relations is proposed in [7]. They calculate the similarity between the two dependency trees of T and H directly. Lexical relation, WordNet and Syntactic Matching for solving the textual entailment problem are used in [8].

The system presented in [9] proposed a novel approach to RTE that exploits a structure-oriented sentence representation followed by a similarity function. The structural features are automatically acquired from tree skeletons that are extracted and generalized from dependency trees.

A syntactic dependency tree approach for the task of textual entailment is used in [10]. This system approach is to construct the syntactic dependency trees for both text and hypothesis sentences and then compare the nodes of the dependency trees by using the semantic similarity between the two nodes. Their approach is closest to method used in the present work. But, a different scoring mechanism and a different set of syntactic relations have been used in the present work. The scoring technique is quite simple and thus easy to compute and interpret.

3 System Description

In this section, we describe our syntactic textual entailment system. The system extracts syntactic structures from the text-hypothesis pairs using Stanford Parser and compares the corresponding structures to determine if the entailment relation is established. The system accepts pairs of text snippets (text and hypothesis) at the input and gives a value at the output: YES if the text entails the hypothesis and NO otherwise. The architecture of the proposed system is described in Figure 1.

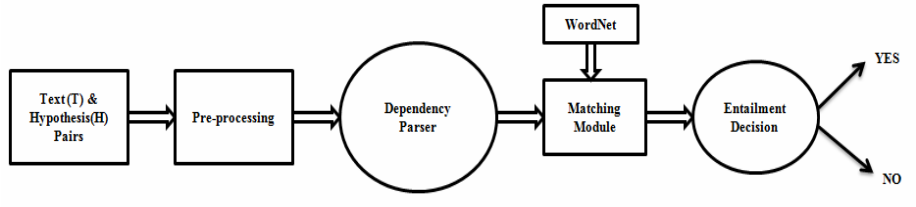

Fig. 1. Syntactic Textual Entailment Recognition System

The various components of the textual entailment recognition system are Pre-processing module, Dependency Parser module, Matching module and Entailment Decision module. Each of these modules is now being described in subsequent subsections.

3.1 Pre-processing Module

The system accepts pairs of text snippets (text and hypothesis) at the input and gives the output: YES if the text entails the hypothesis and NO otherwise. An example text-hypothesis pair from the RTE-3 development set is shown in Figure 2.

<pair id="1" **entailment**="YES" **task**="IE" **length**="short" >
<t>The sale was made to pay Yukos' US$ 27.5 billion tax bill, Yuganskneftegaz was originally sold for US$ 9.4 billion to a little known company Baikalfinansgroup which was later bought by the Russian state-owned oil company Rosneft .**</t>**
<h>Baikalfinansgroup was sold to Rosneft.**</h>**
</pair>

Fig. 2. RTE-3 development set text-hypothesis pair

We replace in all development data the expressions “aren’t” with “are not”, “didn’t” with “did not”, “doesn’t” with “does not”, “won’t” with “will not”, “don’t” with “do not”, “hasn’t” with “has not”, “isn’t” with “is not”, “couldn’t” with “could not”, “ă” with “a”, "á" with “a”, "š" with “s”, "ž" with “z”, "ó" with “o”. These expressions are either abbreviations or include special characters for which the dependency parser gives erroneous results. It has also been observed that escape characters like ", …, ‘ and & are present in the text and in the hypothesis parts and these were removed. All the above pre-processing methods were applied on the development set and the test set.

3.2 Dependency Parser Module

This module is based on the Stanford Parser [11], which normalizes data from the corpus of text and hypothesis pairs, accomplishes the dependency analysis and creates appropriate structures Our Entailment system uses the following features,

a. Subject: The dependency parser generates *nsubj* (nominal subject) and *nsubjpass* (passive nominal subject) tags for the subject feature. Our entailment system uses these tags.
b. Object: The dependency parser generates *dobj* (direct object) as object tags.
c. Verb: Verbs are wrapped with either the subject or the object.
d. Noun: The dependency parser generates *nn* (noun compound modifier) as noun tags.
d. Preposition: Different type of prepositional tags are *prep_in*, *prep_to*, *prep_with* etc. For example, in the sentence “A plane crashes in Italy.”, the prepositional tag identified is *prep_in*(in, Italy).
e. Determiner: Determiner denotes a relation with a noun phase. The dependency parser generates *det* as determiner tags. For example, the parsing of the sentence “A journalist reports on his own murders.” generates the determiner relation as *det*(journalist,A).
f. Number: The numeric modifier of a noun phrase is any number phrase. The dependency parser generates *num* (numeric modifier). For example, the parsing of the sentence “Nigeria seizes 80 tonnes of drugs.” generates the relation *num* (tonnes, 80).

Here is an example from RTE-4 data set. For the sentence, “Nigeria seizes 80 tonnes of drugs”, the Stanford Dependency Parser generates the following set of dependency relations:

[
nsubj(seizes-2, Nigeria-1),
num(tonnes-4, 80-3),
dobj(seizes-2, tonnes-4),
prep_of(tonnes-4, drugs-6)
]

3.3 Matching Module

After dependency relations are identified for both the text and the hypothesis in each pair, the hypothesis relations are compared with the text relations. The different features that are compared are noted below. In all the comparisons, a matching score of 1 is considered when the complete dependency relation along with all of its arguments matches in both the text and the hypothesis. In case of a partial match for a dependency relation, a matching score of 0.5 is assumed.

a. Subject-Verb Comparison: The system compares hypothesis subject and verb with text subject and verb that are identified through the *nsubj* and *nsubjpass* dependency relations. A matching score of 1 is assigned in case of a complete match. Otherwise, the system considers the following matching process.
b. WordNet Based Subject-Verb Comparison: If the corresponding hypothesis and text subjects do match in the subject-verb comparison, but the verbs do not match, then the WordNet distance between the hypothesis and the text is compared. If the value of the WordNet distance is less than 0.5, indicating a closeness of the corresponding verbs, then a match is considered and a matching score of 0.5 is assigned. Otherwise, the subject-subject comparison process is applied.
c. Subject-Subject Comparison: The system compares hypothesis subject with text subject. If a match is found, a score of 0.5 is assigned to the match.
d. Object-Verb Comparison: The system compares hypothesis object and verb with text object and verb that are identified through *dobj* dependency relation. In case of a match, a matching score of 0.5 is assigned.
e. WordNet Based Object-Verb Comparison: The system compares hypothesis object with text object. If a match is found then the verb corresponding to the hypothesis object with text object's verb is compared. If the two verbs do not match then the WordNet distance between the two verbs is calculated. If the value of WordNet distance is below 0.50 then a matching score of 0.5 is assigned.
f. Cross Subject-Object Comparison: The system compares hypothesis subject and verb with text object and verb or hypothesis object and verb with text subject and verb. In case of a match, a matching score of 0.5 is assigned.
g. Number Comparison: The system compares numbers along with units in the hypothesis with similar numbers along with units in the text. Units are first compared and if they match then the corresponding numbers are compared. In case of a match, a matching score of 1 is assigned.
h. Noun Comparison: The system compares hypothesis noun words with text noun words that are identified through *nn* dependency relation. In case of a match, a matching score of 1 is assigned.

i. Prepositional Phrase Comparison: The system compares the prepositional dependency relations in the hypothesis with the corresponding relations in the text and then checks for the noun words that are arguments of the relation. In case of a match, a matching score of 1 is assigned.
j. Determiner Comparison: The system compares the determiner in the hypothesis and in the text that are identified through *det* relation. In case of a match, a matching score of 1 is assigned.
k. Other relation Comparison: Besides the above relations that are compared, all other remaining relations are compared verbatim in the hypothesis and in the text. In case of a match, a matching score of 1 is assigned.

WordNet [12] is one of most important resource. The WordNet 2.0 has been used for WordNet based subject-verb comparison and WordNet based Object-verb comparison. API for WordNet Searching RiWordnet [13] provides Java applications with the ability to retrieve data from the WordNet database.

3.4 Entailment Decision

Each of the matches through the above comparisons is assigned some weight learnt from the development corpus. A threshold of 0.30 has been set on the fraction of matching hypothesis relations based on the development set results that gives optimal precision and recall values for both YES and NO entailment. The threshold score has been applied on the RTE-4 gold standard test set using the same methods of dependency parsing followed by comparisons.

4 Experiments on the Development and the Test Data and the Results

In RTE-4 there was no development set provided, as the pairs proposed were very similar to the ones contained in RTE-3 development and test sets, which could therefore be used to train the systems. Four applications – namely IE, IR, QA and SUM – were considered as settings or contexts for the pair generation. The length of the H's was the same as in the past data sets (RTE-3); however, the T's were generally longer. The RTE-3 development set was used to train our entailment system to identify the threshold values for the various measures towards entailment decision. The 2-way RTE-3 development set consisted of 800 text-hypothesis pairs. The RTE-4 test set consisted of 1000 text-hypothesis pair.

In our textual entailment system, the method was run separately on the RTE-3 development set and two-way entailment (YES or NO) decisions were obtained for each text-hypothesis pair. Experiments were carried out to measure the performance of the final RTE system. It is observed that the precision and recall measures of the final RTE system are best when final entailment decision is based on positive results with threshold value 0.30. The results on the RTE-3 development data set for each task (IE/IR/QA/SUM) are shown in Table 1. It is observed that the system performs best on the development set for the QA task and worst on the development set for the IE task. This points to the requirement of system tuning with respect to the associated

task but this point has not been studied further. Two baseline systems have been developed in the present task. The Baseline–1 system assigns YES tag to all the text-hypothesis pairs and the Baseline–2 system assigns NO tag to all the text0hypothesis pairs. The results obtained on Baseline–1 and Baseline–2 systems on the RTE-3 development data set and the RTE-4 test data set have been shown in Table 2 and Table 3 respectively. The results on the RTE-3 development set for YES and NO entailment decisions are shown in Table 4. The results on RTE-4 test set are shown in Table 5. The system performance on the RTE-3 development set and RTE-4 test set are clearly above the baseline.

Table 1. RTE 3 development set task when threshold value 0.30

RTE 3 Development Set		IE		IR		QA		SUM	
		YES	NO	YES	NO	YES	NO	YES	NO
Cut Off 0.30	Precision	0.55	0.47	0.66	0.65	0.76	0.65	0.68	0.58
	Recall	0.65	0.38	0.48	0.80	0.64	0.77	0.57	0.69

Table 2. Baseline-1 system for RTE-3 Development Set and RTE-4 Test Set

	Entailment Decision	No. of Entailment in Gold standard	Baseline-1	Precision
RTE-3 Development Set	YES	412	800	51.50%
	NO	388	0	0%
RTE-4 Test Set	YES	500	1000	50.00%
	NO	500	0	0%

Table 3. Baseline-2 system for RTE-3 Development Set and RTE-4 Test Set

	Entailment Decision	No. of Entailment in Gold standard	Baseline-2	Precision
RTE-3 Development Set	YES	412	0	0%
	NO	388	800	48.50%
RTE-4 Test Set	YES	500	0	0%
	NO	500	1000	50.00%

Table 4. RTE 3 development set when threshold value 0.30

Entailment Decision	No. of Entailment in Gold standard	No. of correct Entailment in our system	Total No. of Entailment given by our system	Precision	Recall
YES	412	244	371	65.76%	59.22%
NO	388	261	429	60.83%	67.26%
Overall	800	505	800	63.12%	63.12%

Table 5. RTE 4 test set when threshold value 0.30

Entailment Decision	No. of Entailment in Gold standard	No. of correct Entailment in our system	Total No. of Entailment given by our system	Precision	Recall
YES	500	265	484	54.75%	53.00%
NO	500	281	516	54.45%	56.20%
Overall	1000	546	1000	54.60%	54.60%

5 Conclusions

Results show that a syntactic-based approach is not enough to tackle appropriately the textual entailment problem. Experiments have been started for a semantic based RTE task. In the present task, the final RTE system has been optimized for the entailment YES/NO decision using the development set. The role of the application setting for the RTE task has also not been looked into. This needs to be experimented in future. Finally, the two way task has to be upgraded to the three way task.

References

[1] Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the First PASCAL Recognizing Textual Entailment Workshop (2005)

[2] Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The Second PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy (2006)

[3] Voorhees, E.M., Harman, D.: Overview of the seventh text retrieval conference. In: Proceedings of the Seventh Text REtrieval Conference (TREC-7). NIST Special Publication (1999)

[4] Giampiccolo, D., Dang, H.T., Magnini, B., Dagan, I., Cabrio, E.: The Fourth PASCAL Recognizing Textual Entailment Challenge. In: TAC 2008 Proceedings (2008), http://www.nist.gov/tac/publications/2008/papers.html
[5] Vanderwende, L., Coughlin, D., Dolan, B.: What syntax can contribute in entailment task. In: Proceedings of the First PASCAL Recognizing Textual Entailment Workshop, pp. 13–16 (2005)
[6] Herrera, J., Peñas, A., Verdejo, F.: Textual Entailment Recognition Based on Dependency Analysis and WordNet. In: Proceedings of the First PASCAL Recognizing Textual Entailment Workshop, pp. 21–24 (2005)
[7] Kouylekov, M., Magnini, B.: Tree Edit Distance for Recognizing Textual Entailment: Estimating the Cost of Insertion. In: Proc. of the PASCAL RTE-2 Challenge, pp. 68–73 (2006)
[8] Blake, C.: The Role of Sentence Structure in Recognizing Textual Entailment. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 101–106 (2007)
[9] Wang, R., Neumann, G.: Recognizing Textual Entailment Using Sentence Similarity based on Dependency Tree Skeletons. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 36–41 (2007)
[10] Varma, V., Krishna, S., Garapati, H., Reddy, K., Pingali, P., Ganesh, S., Gopisetty, H., Bysani, P., Katragadda, R., Sarvabhotla, K., Reddy, V.B., Bharadwaj, R.: Recognizing Textual Entailment (RTE) Track. In: Text analysis conference 2008 Proceedings (2008)
[11] Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: ACL 2003, pp. 423–430 (2003)
[12] Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
[13] RiWordnet API Tool, http://www.rednoise.org/rita/wordnet/documentation/index.htm