Recognizing Textual Entailment Using a Machine Learning Approach

Miguel Angel Ríos Gaona[1], Alexander Gelbukh[1], and Sivaji Bandyopadhyay[2]

[1] Center for Computing Research, National Polytechnic Institute, Mexico
mriosb08@sagitario.cic.ipn.mx, gelbukh@gelbukh.com
[2] Computer Science & Engineering Department, Jadavpur University, Kolkata 700 032 India
sivaji_cse_ju@yahoo.com

Abstract. We present our experiments on Recognizing Textual Entailment based on modeling the entailment relation as a classification problem. As features used to classify the entailment pairs we use a symmetric similarity measure and a non-symmetric similarity measure. Our system achieved an accuracy of 66% on the RTE-3 development dataset (with 10-fold cross validation) and accuracy of 63% on the RTE-3 test dataset.

Keywords. Recognizing Textual Entailment, text similarity measures, non-symmetric measures.

1 Introduction

One of the largest challenges in Natural Language Processing (NLP) is to provide a computer with the linguistic knowledge necessary to successfully perform language-oriented tasks. For example, for the query "*What does Peugeot manufacture?*" a Question Answering (QA) system must be able to recognize, or infer, and answer which may be expressed differently from the query. For example, from a text "*Chrétien visited Peugeot's newly renovated car factory*" the system should be able to infer a hypothesized answer from "*Peugeot manufactures cars*". A fundamental phenomenon in NLP is the variability of a semantic expression: the same meaning can be expressed in, or inferred from, different text.

A task that addresses this inference phenomenon is Recognizing Textual Entailment (RTE). Textual Entailment is defined as a directed relationship between pairs of text expressions, denoted by T (text) and H (hypothesis). We say that T entails H if the meaning of H can be inferred from the meaning of T as could typically be interpreted by people [3].

Moreover, many NLP tasks have strong relationship to entailment: in summarization, a summary should be entailed by the text; paraphrases can be seen as mutual entailment between a text T and a hypothesis H; in Information Extraction (IE), the extracted information should also be entailed by the text; in Question Answering (QA) and Information Retrieval (IR), the answer obtained for a query must be entailed by the supporting snippet of text.

To address the RTE task, different methods have been proposed, with varying degree of success. These methods can be classified by the type of representation of the

entailment pair. The commonly used criteria for entailment recognition are similarity measures between T and H, the coverage of H by T in lexical representation methods and lexical-syntactic representation methods, and the ability to infer H from T, in the logical representation approach. Some authors [8] try to detect non-entailment, by looking for various kinds of mismatch between the text and the hypothesis.

In this paper, we propose the use of a symmetric similarity measure and a non-symmetric similarity measure as features in a Machine Learning (ML) algorithm for RTE. The symmetric measure is the cosine string similarity measure. The non-symmetric measure is given by measuring the causal relation between the entailment pairs. This measure uses the relative frequencies of words in a cause-effect set. The cause-effect set is created by retrieving sentences from the Web that contain the discourse marker *because*.

The hypothesis behind our system is that the symmetric similarity measures can not answer correctly (cover) all the entailment pairs, and with the addition of the non-symmetric similarity measures the remaining pairs might be covered.

The paper is structured as follows. In Section 2, we present an overview of related work. In Section 3, we describe the measures used in our experiments. In Section 4, we give the experimental results and the comparison with the state of the art. Finally, Section 5 concludes the paper.

2 Related Work

The RTE approaches can be classified by the textual entailment phenomena they address or by type of linguistic representation (*levels of language*) of the entailment pair they use. Each type of linguistic representation requires its own operations in order to establish the entailment decision, e.g., word matching at the lexical level, tree edit distance at the syntactic level, etc.

In some systems, the entailment decision ("T entails H") is made by comparing the score of the given operation with a threshold learned from an annotated corpus: If this score is greater than the threshold, the system answers "true", otherwise the answer is "false". There are different techniques to learn a threshold.

The main operations on a linguistic representation are similarity measures. Most of these similarity measures are symmetric. However, a symmetric measure can not capture important aspects in the T → H (T implies H) relation. For example, if we alter the entailment relation (i.e., H → T) a symmetric function will give the same score. Therefore, some authors, e.g., [14], propose a non-symmetric similarity measure. Such measures have been used in RTE-1 Challenge.

Glickman [5] defines the entailment relation as follows: T entails H if $P(H|T) > P(H)$. The probabilities are calculated on the base of the Web. The accuracy of this system is the best for RTE-1 (56%).

Another non-symmetric method was proposed by Kouylekov [9], who uses the definition: T entails H if there exists a sequence of transformations applied to T such that H is obtained, with a total cost below of a certain threshold. The following transformations are allowed: insertion: insert a node from the dependency tree of H into the dependency tree of T; deletion: delete a node from the dependency tree of T; sub-

stitution: change a node in the T for a node of H. Each transformation has a cost and the cost of edit distance between T and H, ed(T, H) is the sum of costs of all applied transformations. The entailment score of a given pair is calculated as

$$\text{score}(T, H) = \text{ed}(T, H),$$

If this score is below a learned threshold, the relation T → H holds. The accuracy of this method is also of 56%.

In [14], an even "more non-symmetric" measure is proposed: when the edit distance (which was a modified Levenshtein distance) fulfills the relation:

$$\text{ed}(T,H) < \text{ed}(H,T),$$

then the relation T → H holds.

Other authors use a definition that in terms of representation of knowledge as feature structures could be formulated as: T entails H if H subsumes T [14]. The method used in [3] is also non-symmetric: T entails H if H is not informative in respect to T.

A method of establishing the entailment relation could be obtained using a non-symmetric measure of similarity between two texts presented by Corley and Mihalcea [2], who define the similarity between the texts T_i and T_j with respect to T_i as:

$$sim(T_i, T_j)_{Ti} = \frac{\sum_{pos}\left(\sum_{wk \in ws^{ti}_{pos}} (\max Sim(w_k) \times idf(w_k))\right)}{\sum_{pos}\sum_{wk \in ws^{ti}_{pos}} idf(w_k)}$$

Here the sets of open-class words (nouns, verbs, adjective, and adverbs) in each text segment are denoted by the PoS (part of speech) of WST_i and the PoS of WST_j. For a word w_k with a given PoS in T_i, the highest similarity of the words with the same PoS in the other text T_j is denoted by maxSim(w_k).

Basing on this text-to-text similarity metric, we derive a textual entailment recognition system by applying the lexical refutation theory [14]. As the hypothesis H is less informative than the text T, for a TRUE pair the following relation will hold:

$$\text{sim}(T, H) \times T < \text{sim}(T, H) \times H.$$

This relation can be proved using lexical refutation. A general scheme of the solution is the follows: to prove T → H it is necessary to prove that the set of formulas {T; neg-H} is lexically contradictory (T and negH also denote the sets of disjunctive clauses of T and negH).

3 Similarity Measures Used in the Experiments

Many systems for RTE are based on similarity measures; we used these measures to train a machine learning algorithm. The entailment decision is given by a classifier, where the classes are "true" and "false".

We will now describe the two string similarity measures used in our experiments. We chose two measures as features: the cosine symmetric measure and the causal non-symmetric measure.

3.1 Cosine Similarity Measure

Large classes of measures of semantic similarity are best conceptualized as measures of vector similarity. We consider binary vectors, that is, vectors with entries that are either 0 or 1. The simplest way to describe a binary vector is as the set of its nonzero values.

Cosine similarity is a measure of similarity between two *n*-dimensional vectors obtained by finding the cosine of the angle between them. It is often used to compare documents in text mining. In addition, it is used to measure cohesion within clusters in data mining. Cosine similarity is also widely used in information retrieval to calculate the similarity between documents or sentences. Given two vectors of attributes, A and B, the cosine similarity θ is calculated using the dot product and magnitude as:

$$COS(A,B) = \frac{|A \cap B|}{\sqrt{|A| \times |B|}}.$$

Note that this is a symmetric measure, that is, COS(A, B) = COS(B, A).

3.2 Causal Non-symmetric Measure

First, we give a brief theoretical introduction to the measure. A causal relation refers to the relation between a cause and its effect or between regularly correlated events. The type of coherence relation we used is cause-effect is illustrated below. In the example below, (1) states the cause for the effect given in (2):

1. *There was bad weather at the airport*
2. *Our flight was delayed*.

The causal relation subsumes the cause and the explanation relations discussed by Hobbs [7]. Hobbs's causal relation holds if a discourse segment stating a cause occurs before a discourse segment stating an effect; an explanation relation holds if a discourse segment stating an effect occurs before a discourse segment stating a cause. The causal relation is encoded by adding a direction. In a graph, this can be represented by a directed arc going from cause to effect.

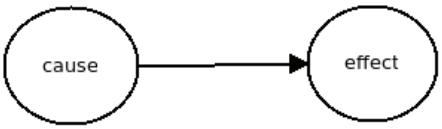

Figure 1. Cause-effect graph.

In Figure 1, the causality is a directional relationship, in the same way as the relationship between the members of an entailment pair. A non-symmetric similarity measure based on the count of co-occurrences of causal lexical pairs could be as follows: If a word x is a necessary (likely) cause of a word y, then the presence of y necessarily (likely) implies the presence of x.

In [11], a non-symmetric similarity measure is proposed based on the treatment the entailment pair as a causal relation, where the text T is a cause and the hypothesis H is its effect, i.e., T causes H. The non-symmetric similarity measure is based on the count of co-occurrences of causal lexical pairs from cause-effect (C-E) pairs extracted from a corpus.

Algorithm 1. The non-symmetric similarity measure.

```
For each word t_i in T
    For each word h_j in H
        ce_j = causal frequency(t_i, h_j)
        e_j = causal frequency(h_j)
    max_i = argmax(ce_j / e_j)
nonsymmetric(T,H) = • max_i
```

In Algorithm 1 used for our experiments, the first causal frequency function is the count of words t_i and h_i related by a cue phrase (for example, a sentence "H … *because* … T") in a corpus of C-E pairs, and the second causal frequency function is the count of word h_i in the C-E pairs. This gives a non-symmetric score, because the frequency counts of "T causes H" is not the same as "H causes T".

4 Experimental Results

In this section we first describe the linguistic processing for feature extraction and then the experiments over various Machine Learning algorithms. Finally, we give a comparison with the state of the art.

4.1 Experimental Setting

The linguistic processing we used with each entailment pair is as follows:

1. *Tokenizing*. As usually, the first step of processing is to divide the input text into units called tokens. Each of them is a word, a number, a punctuation mark, etc. The treatment of punctuation marks can vary in such process. Our system just strips the punctuation marks out. We consider as word any string between whitespaces and punctuation characters. The whitespace is the main clue used in English texts (RTE benchmark is in English).
2. *Removal of stop words*. The system removes any stops words that are listed in the corresponding list, such as *the*, *from*, or *could*. These words have important semantic function in English, but they rarely contribute information if the criterion is a simple word-by-word match.

3. *Measuring similarity*. Similarity measures are applied to each entailment pair, to extract the train and test sets for the machine learning algorithm.

The data we used to collect the frequency of the causal lexical pairs for the causal non-symmetric measure was from training sentences which contain the cue word *because*. The causal sentences were separated in two parts: one corresponding to the cause and the other one corresponding to its effect, to finally form the cause-effect pairs. The sentences were extracted from the Sketch Engine system over a large corpus (ukWAC from the Sketch Engine[1]). The Sketch Engine is a corpus query system that allows the user to view word sketches, thesaurally similar words, and so-called "sketch differences", similarly to the usual Corpus Query Systems (CQS).

4.2 Machine Learning Experiments

The RTE-3 Challenge provided two datasets (a development dataset and a test dataset), each one consisting of 800 entailment pairs. In both datasets, pairs are annotated according to the task. In RTE-2 the length annotation is introduced, with values of either "long" or "short." In addition, the development sets are annotated as to whether each pair is in the entailment relation or not.

We applied the linguistic preprocessing to each RTE-3 dataset; the result is a set of vectors of two features. These sets are used to train and test a classifier. We used the WEKA[2] machine learning platform [15] for our experiments.

We ran several experiments with various machine learning algorithms, including Support Vector Machine, AdaBoost, Naïve Bayes, among others. We used the RTE-3 development dataset to train the classifiers. The results of the 10 fold-cross validation are show in Table 1.

The Support Vector Machine (SVM) and the Naïve Bayes achieved the best results in the experiments during the training phase. Then we used these two algorithms to perform the classification over the RTE-3 test dataset.

Table 1. 10 fold-cross validation results over the RTE-3 development dataset.

Algorithm	Accuracy
SVM	**66.37%**
NaïveBayes	**65.87%**
AdaBoost	65.25%
BayesNet	65.25%
LogitBoost	65%
MultiBoostAB	64.125%
RBFNetwork	64.87%
VotedPerceptron	51.75%

[1] http://www.sketchengine.co.uk/

[2] WEKA. http://www.cs.waikato.ac.nz/ml/weka/

The SVM algorithm tries to compute the hyperplane that best separates the set of training examples (the hyperplane with maximum margin). On the other hand, the Naïve Bayes algorithm is a classification algorithm based on the Bayes rule that assumes the features are all conditionally independent from one another. The value of this assumption is that it dramatically simplifies the representation of the probability $P(X \mid Y)$ and the problem of estimating it from the training data.

4.3 Comparison with Previous Results

The experimental results are summarized in Table 2. We compare our system against other Machine Learning systems which use features based on similarity measures. All the systems were tested over the RTE-3 test dataset. In Table 2 we report the results with the Naïve Bayes. The SVM achieved an accuracy of 62.87%.

Table 2. Comparison with previous results.

System	Number of Features	Accuracy
Our system with Naïve Bayes	**2**	**63.5%**
Li et al. (2007)	7	62.75%
Malakasiotis and Androutsopoulos (2007)	10	61.75%
Ferrés and Rodriguez (2007)	12	61.50%

Therefore, our system outperformed the other machine learning systems, which used more features. Indeed, we used only two features (one symmetric and one non-symmetric similarity measure), while, for example, in [11] the authors used 10 different similarity measures (e.g. Levenshtein distance, Jaro-Winkler, Soundex, etc.).

The approach of Ferrés and Rodriguez [5] for computing distance measures between sentences is based on the degree of overlapping between the semantic content of the two sentences. Obtaining the semantic content implies deep linguistic processing. Upon this semantic representation of the sentences, several distance measures are computed.

Li *et al*. [10] produced seven features for each entailment pair: lexical semantic similarity, named entities, dependent content word pairs, average distance, negation, task, and text length. The last two features are extracted from each pair itself, while others are based on the results of language analyzers.

Finally, the system of Malakasiotis and Androutsopoulos [11] uses SVM's to determine whether each T–H pair constitutes a correct textual entailment pair. In particular, it employs four SVMs, each trained on the development dataset of the corresponding RTE subtask (QA, IR, IE, SUM) and used on the corresponding test dataset. Preliminary experiments indicated that training a single SVM on all four subsets leads to worse results, despite the increased size of the training set, presumably because of differences in how the pairs were constructed in each subtask, which do not allow a single SVM to generalize well over all four. Their system is based on the assumption that string similarity at the lexical and shallow syntactic level can be used to identify textual entailment.

Thus, many ML systems need a complex linguistic processing in order to extract features for modeling the entailment recognition.

5 Conclusions and Future Work

We proposed combined use of symmetric similarity measure and non-symmetric similarity measure as features for a machine learning approach. We have shown that our system outperforms other machine learning approaches to RTE. Furthermore, we have shown that the use of two different types of measures improves the performance of a machine learning system. Finally, our system has the advantage of simplicity and the use of a very limited feature set.

Our system also has competitive accuracy, because the average accuracy for the RTE-3 is about 61%. The state-of the-art (shown by non-machine learning-based systems) for the RTE-3 is about 80%.

In the future we plan to use other non-symmetric similarity measures, i.e., Corley and Mihalcea, Glickman. We will use syntactic and semantic measures (WordNet-based similarity measures) to achieve better performance. In particular, we plan to test deeper semantic processing, including determining and using verb valencies [1]. Finally, we will test our system over the past RTE Challenge datasets as new test and training sets.

Acknowledgements. The work was done under partial support of Mexican Government (SNI, CONACYT grant 50206-H, CONACYT scholarship for Sabbatical stay at Waseda U., COFAA-IPN, and SIP-IPN grant 20100773) and Government of India (CONACYT-DST India funded project).

References

1. Castro-Sánchez, N. A., and Sidorov, G. Analysis of Definitions of Verbs in an Explanatory Dictionary for Automatic Extraction of Actants based on Detection of Patterns. Lecture Notes in Computer Science, N 6177, pp 233–239, 2010.
2. Corley, C., and R. Mihalcea. (2005). Measuring the semantic similarity of texts. Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, Ann Arbor.
3. Dagan, I., and O. Glickman. (2004). Probabilistic textual entailment: Generic applied modeling of language variability. PASCAL workshop on Text Understanding and Mining Monz, C., and M. de Rijke. (2001).Light-Weight Entailment Checking for Computational Semantic. In: P. Blackburn and M. Kohlhase, editors, Proceedings ICoS-3.
4. De Salvo Braz, R., R. Girju, V. Punyakanok, and D. M. Frentiu. (2005). An Inference Model for Word Sense Disambiguation. In Proceedings of KEPT2007, Knowledge Engineering Principles and Techniques, Vol I, Workshop on Recognising Textual Entailment.
5. Ferrés, D., and H. Rodríguez. (2007). Machine Learning with Semantic-Based Distances Between Sentences for Textual Entailment. In Proceedings of the Third Challenge Workshop Recognising Textual Entailment, Prague, Czech Republic.
6. Glickman, O., I. Dagan and M. Koppel. (2005). Web Based Probabilistic Textual Entailment. In Proceedings of the PASCAL Challenges Work-shop on Recognising Textual Entailment.
7. Hobbs, J. R. (1985). Ontological promiscuity. Proceedings of the 23rd annual meeting on Association for Computational Linguistics.

8. Inkpen, D., D. Kipp, and V. Nastase. (2006). Machine Learning Experiments for Textual Entailment. In Proceedings of the Second Challenge Workshop Recognising Textual Entailment, Pages 17-20, 10 April, 2006, Venice, Italy.
9. Kouylekov, M., and B. Magnini. (2006). Tree Edit Distance for Recognizing Textual Entailment: Estimating the Cost of Insertion. In Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice,Italy.
10. Li, B., J. Irwin, E. V. Garcia, and A. Ram. (2007). Machine Learning Based Semantic Inference: Experiments and Observations at RTE-3. In Proceedings of the Third Challenge Workshop Recognising Textual Entailment, Prague, Czech Republic.
11. Malakasiotis, P., and I. Androutsopoulos. (2007). Learning Textual Entailment using SVMs and String Similarity Measures. In Proceedings of the Third Challenge Workshop Recognising Textual Entailment, Prague, Czech Republic.
12. Pérez, D. and E. Alfonseca. (2005). Application of the Bleu algorithm for recognising textual entailments. In Proceedings of the First Challenge Workshop Recognising Textual Etailment, Southampton, U.K.
13. Ríos, M., A. Gelbukh and S. Bandyopadhyay. (2010). Recognizing Textual Entailment with Statistical Methods. MCPR 2010, 2nd Mexican Conference on Pattern Recognition (to be published).
14. Tatar, D.; S. Gabriela; M. Andreea-Diana, and M. Rada. Textual Entailment as a Directional Relation. (2009). Journal of Research and Practice in Information Technology.
15. Witten, H., and E. Frank. (2005). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco.