Journal of Information & Computational Science 9: 1 (2012) 1–8
Available at http://www.joics.com

Applying CRFs and SVMs to Textual Entailment Recognizing [⋆]

Yongmei Tan [a,*], Eduard Hovy [b]

[a] *School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*

[b] *Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, U.S.A.*

Abstract

Textual entailment is knowledge that may prove useful for a variety of applications dealing with inferencing over sentences described in natural language. This paper proposes a combined method to recognize textual entailment. Firstly to calculate the similarity between sentence pair (text T and hypothesis H) based on the TF-IDF values. Secondly to discard the sentence pairs that have high similarity score and are actually not entailed using additional information like features from words, chunks and predicate-argument structures by machine learning methods. In contrast to previous approaches that makes use of all kinds of features. The experimental results show that our method is effective for recognizing textual entailment task. Our experiments also show that the different techniques that we employ perform very differently on some of the subsets of the recognizing textual entailment corpus and as a result, it is useful to use the nature of the dataset as a feature.

Keywords: Textual Entailment; Support Vector Machines; Conditional Random Fields

1 Introduction

Entailment knowledge is an important component for any knowledge-based system, i.e. a system equipped with background information about concepts and relations between them that are characteristic of the domain in which the system is deployed.

The Recognizing Textual Entailment (RTE) task consists of developing a system that, given two text fragments, can determine whether the meaning of one text is entailed i.e. can be inferred, from the other text [1].

The RTE challenge examples are drawn from multiple domains, providing a relatively task-neutral setting in which to evaluate contributions of different component solutions, and RTE researchers have already made incremental progress by identifying sub-problems of entailment, and developing ad-hoc solutions for them.

The challenge of the task is threefold. First, the texts and hypotheses are not modified as compared to the original source, so they may contain incomplete sentences, spelling errors, grammar

[⋆]Project supported by the Central Universities and the National Natural Science Foundation of China (NSFC) (90920006, 61273365).

[*]Corresponding author. Tel.: +86-10-62283259
Email address: ymtan@bupt.edu.cn (Yongmei Tan).

January 2012

errors and abbreviations, etc. Second, texts and hypotheses are interpreted within the context of the topic, as they rely on explicit and implicit references to entities, dates, places, events, etc. pertaining to the corpus [2]. Third, there are much more negative pairs than positive pairs, as for RTE7 development set there are totally 21420 candidate pairs, while 1136 positive pairs (entailing) and 20284 negative pairs (not entailing). RTE6 development set there are totally 15955 candidate pairs, while the 897 positive pairs.

We focus on the similarity estimated as the degree of word overlap between text and hypothesis based on the intuition that entailment is related to the similarity between text and hypothesis and then filter the TH pairs that have high similarity score and are actually not entailment.

2 Related Work

2.1 Recognizing Textual Entailment (RTE)

Textual entailment lies at the heart of many NLP problems. A database encoding entailment can serve as a useful tool for the recognition of information implicitly present in a text. A number of studies [3, 4] have shown that entailment knowledge has a significant positive effect on the performance of QA systems by making it possible to retrieve not only explicit, but also implicit answers to a question.

Recognizing entailment between lexical expressions is an important subproblem of textual entailment recognition, the task of establishing entailment between two larger text fragments. Textual entailers have been used to achieve a considerable boost in performance of QA [5] and document retrieval systems [6]. Similar benefits can be expected in related tasks such as Text Categorization and Information Extraction, where the resource can be used for term and query expansion. In Document Summarization, it can help to determine sentences that have the same meaning and thus avoid repetitive content in the summary. Textual entailment can also improve automatic evaluation of Machine Translation by recognizing similar content present in different target variants of the same source text.

The textual entailment is defined as a directional relationship between two text fragments - T, the entailing text and H, the entailed text - so that T entails H if, typically, a human reading T would infer that H is most likely true [7].

This definition of entailment is based on common human understanding of language as well as background knowledge; in fact, for textual entailment to hold it is required that text and knowledge entail hypothesis, but knowledge alone cannot entail hypothesis [8]. In other words, hypothesis is not entailed if hypothesis is true regardless of text. TF-IDF is a weight often used in information retrieval and text mining, and can be found in previous RTE paper. TF-IDF is a numerical statistic which reflects how important a word is to a document in a collection or corpus and is used by previous RTE system.

2.2 Support Vector Machines (SVMs)

SVM is a relatively new machine learning technique first presented by Vapnik [9, 10]. Based on the structural risk minimization principle of computational learning theory, SVMs seek a decision boundary to separate training examples into two classes and to make decisions based on the support vectors which are selected as the only effective examples in the training set.

Like other inductive learning approaches, SVMs take a set of training examples (e.g., feature vectors with binary values) as input, and find a classification function which maps them to classes. In this paper, a separating hyper-plane described by a weight vector $\vec{w}$ and a threshold b perfectly divides the training data $\vec{x}$ into 2 classes labeled as $y \in \{-1, +1\}$, each side containing data examples with the same class label only.

$$(\vec{w} \cdot \vec{x}) + b = 0 \quad \vec{w} \in R^n, b \in R \tag{1}$$

Then SVMs learn linear decision rules:

$$y(\vec{x}) = sign(g(\vec{x})) = sign(\vec{w} \cdot \vec{x} + b) = \begin{cases} +1 & if \vec{w} \cdot \vec{x} + b > 0 \\ -1 & otherwise \end{cases} \tag{2}$$

The idea of structural risk minimization is to find a hypothesis for which one can guarantee the lowest probability bound for generalization error. This can be achieved by finding the optimal hyper-plane, i.e., the hyper-plane with the maximum margin. Margin is defined as the distance between the hyper-plane and the training samples which are most close to the hyper-plane.

Maximizing the margin means that the closest samples (support vectors) exist on both sides of the separating hyper-plane and the hyper-plane lies exactly in the middle of these support vectors. Computing this optimal hyper-plane is equivalent to solving the following optimization problem.

$$\begin{gathered} Minimize \quad \tfrac{1}{2} \parallel w \parallel^2 \\ subject\ to: y_i[(w \cdot x_i) - b] \geq 1 \quad (i = 1, \ldots, l) \end{gathered} \tag{3}$$

2.3 Conditional Random Fields (CRFs)

CRFs are a recently introduced from of conditional model that allow the strong independence assumptions of HMMs to be relaxed, as well as overcoming the label-bias problem exhibited by MEMMs [11, 12]. This allows the specification of a single joint probability distribution over the entire label sequence given the observation sequence, rather than defining per-state distributions over the next states given the current state. The conditional nature of the distribution over label sequences allows CRFs to model real-world data in which the conditional probability of a label sequence can depend on non-independent, interacting features of the observation sequence. In addition to this, the exponential nature of the distribution chosen by Lafferty et al. enables features of different states to be traded off against each other, weighting some states in a sequence as being more important than others.

CRFs are defined as follows. Let $X = x_1x_2 \ldots x_T$ denote some observed input data sequences, such as a sequence of words in training data. Let $Y = y_1y_2 \ldots y_T$ be a set of finite state machine (FSM) states, each of which is associated with a label. By the Hammersley-Clifford theorem, CRFs define the conditional probability of a state sequence given an input sequence X

$$P(Y|X) = \frac{1}{Z_x} exp(\sum\nolimits_{i=1}^{T} \sum\nolimits_{k} \lambda_k f_k(y_{i-1}, y_i, X, t)) \tag{4}$$

where Z_X is a normalization factor over all candidate paths. In other words, it is the sum of the "scores" of all possible state sequence.

$$Z_x = \sum\nolimits_{y \in Y} exp(\sum\nolimits_{i=1}^{T} \sum_{k} \lambda_k f_k(y_{i-1}, y_i, X, t)) \tag{5}$$

$f_k(y_{i-1}, y_i, X, t)$is a feature function. The feature functions can measure any aspect of a state transition $y_{t-1} \rightarrow y_t$, and the observation sequence X , centered at the current time step t .

$\lambda_k$is a learned weight associated with feature f_k . Large positive values for f_k indicate a preference for such an event, while large negative values make the event unlikely.

Given such a model as defined in Equ. (4), the most probable labeling sequence for an input X is Y^* which maximizes a posterior probability.

$$Y^* = argmax_Y P_\lambda(Y|X) \tag{6}$$

It can be found with dynamic programming using the Viterbi algorithm.

In the case of the commonly used graph structure for modeling sequential data, the general form of Equ. (4) can be expanded to

$$P(Y|X) = \frac{1}{Z_X} exp(\sum_{i=1}^{T} \sum_k \lambda_k f_k(y_{i-1}, y_i, X) + \sum_{i=1}^{T} \sum_k \mu_k g_k(y_i, X)) \tag{7}$$

Where each $f_k(y_(i-1), y_i, X)$ is a feature of the entire observation sequence and the labels at position i and $i-1$ in the corresponding label sequence, each $g_k(y_i, X)$ is a feature of the label at position i and the observation sequence, and λ_k and μ_k are feature weights.

In this situation, the parameters λ_k and μ_k corresponding to these features are equivalent to the algorithm of HMMs transition and emission probabilities. Although it encompasses HMM-like models, the class of CRFs is much more expressive, because it allows arbitrary dependencies on the observation sequence [11].

3 System Architecture

Our system designed to recognize textual entailment typically employs lexical information and removes the false TH pairs that have high similarity score and are actually not entailment. As Fig. 1 illustrated, the system architecture can be divided into two steps.

The first step chooses the positive pairs based on the similarity estimated as the degree of word overlap between text and hypothesis. For the most positive pairs, the similarity values are high and for the most negative pairs, the similarity values are low, we choose the positive pairs by threshold.

Using additional information like features from words, chunks and predicate-argument structures, the second step removes the false positive pairs among all positive pairs generated in the previous step.

3.1 Step 1: Choosing

Because there are a lot of noises in the data set, we do processing to improve the quality of the text and hypothesis pairs. The tag *'Q:'* and *'A'* within text is deleted. Uppercase is converted to lowercase in order to improve the performance of word overlapping. We replace *"hasn't"* with *"has not"*, *"isn't"* with *"is not"* within text, and etc.

Our algorithm calculates words overlapping and builds the comparison on the basis of words, so that we use the TreeTagger tool[1] to do part-of-speech (POS) tagging and stemming, with a higher degree of precision. We replace *'¡unkonwn¿'* with the initial word and replace *'@card@'* with the initial number and fixes the bug that TreeTagger produces many '\ *t*' in the output file.

Stopwords usually are high-frequency words like the, to, and etc. They have little lexical content and their presence in a text fails to distinguish it from other texts. We remove all words of length 1 based on the experimental results of RTE6 data set.

Coreference identification is based on named entity recognition, so that LingPipe extracts mentions of people, companies, locations, organizations, and etc., and then to identify coreference,

[1] http://www.cele.nottingham.ac.up/ ccztk/treetagger.php

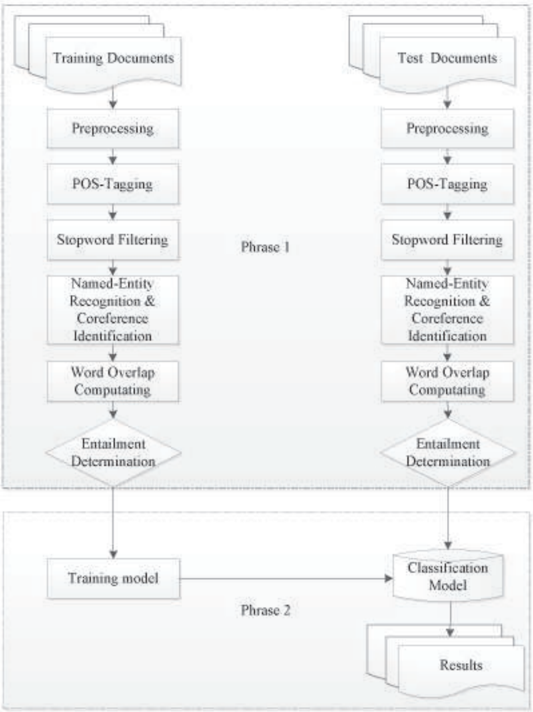

Fig. 1: System Architecture

which can give ids to certain noun phrases and pronouns, the same id indicates the same identity. We do some post-processing to correct some errors within the result files.

In the standard TF-IDF algorithm, TF measures the term frequency in a document, we simply define our TF is equal to 1.0 because we use the intersection of the text and hypothesis pairs to calculate word overlap, and every word of a set must be unique. Our IDF formula is as the same as the standard one. We use all the unique sentences (including text and hypothesis) in the given text corpus to train the TF-IDF model, especially, if a test pair includes a word that never appeared in the given corpus, we consider it as a rare word and give it the maximum value of all the appeared words. Moreover, we think TF-IDF weight can cover the effect by using stop list, which merely consider several common words as zero-weighted.

A higher degree of matching between text and hypothesis has been taken as indication of a semantic relation. We first get the word set of hypothesis and the intersection of text and hypothesis, and then use the TF-IDF weighted set of the intersection divide the weighted set of hypothesis, to get the overlap score of candidate pair. Finally we use the threshold trained with previous corpus as a criterion to compare with the overlap score. *TH* pairs with score higher than the threshold will be marked as true; and candidate pairs with score lower than the threshold will be marked as false.

3.2 Step 2: Classification

Firstly we use LIBSVM to train the classification model, which is a popular library for SVMs [2] [13]. Secondly the model for classification is trained by MALLET [14], which is a famous CRFs package [3] [11]. The conditional modelling approaches have shown advantages at modelling

[2] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[3] http://mallet.cs.umass.edu/

Table 1: Feature Description

Feature	Description
Lexical feature	overlap score of TH pair without POS overlap score of TH pair with POS cosine similarity of TH pair
syntactic feature	the matching ratios for the same grammatical relations (e.g., nn, abbrev, nsubj) and the same governor in TH pair the number of negation mismatch in TH pair

overlapping textual features. For SVMs and CRFs models, the sorts of feature are shown in Table 1.

4 Results and Discussion

4.1 Experimental Data

The RTE7 data set is composed of 20 topics, 10 used for the Development Set and 10 for the Test Set. For each topic, the RTE7 Main Task data consist of:

a. A number of Hypotheses (between 20 and 40) referring to the topic. The Hypotheses are standalone sentences taken from the TAC Update Summarization corpus.

b. A set of 10 documents, corresponding to the *Cluster A* corpus.

c. For each H, a list of up to 100 candidate entailing sentences (the Ts) from the *Cluster A* corpus, together with their location in the corpus and Lucene ranking score.

While Ts are naturally occurring sentences in a corpus and are to be taken as they are, the Hs were slightly modified from the originals so as to make them standalone sentences [8].

The RTE7 data set is composed of 20 topics, 10 used for the development set and 10 for the test set. The development set is composed of 100 documents and contains globally 284 hypotheses. The test set is also composed of 100 documents and contains globally 272 hypotheses. There are much more negative pairs than positive pairs.

System results are compared to a human-annotated gold standard and the metrics used to evaluate system performances are Precision, Recall, and F-measure.

4.2 Metrics

System results were compared to a human-annotated gold standard and the metrics used to evaluate system performances were Precision, Recall, and F-measure.

The official metric chosen for ranking systems was micro-averaged F-measure. Additionally, macro-averaged results for topics were made available to participants. As systems were not forced to retrieve at least one entailing sentence for each topic, in order to calculate macro-averaged results it was decided that, if no sentence was returned for a given topic, the Precision for that topic is 0. Also, as many Hs had no entailing sentences, macro-averaged results for hypotheses were not calculated [8].

4.3 Experimental Results

Our system uses a threshold to judge whether the hypothesis can be entailed from the relative text or not and then filter the TH pairs that have high similarity score and are actually not entailment.

Table 2: Main task results for RTE7 test set

Run	Micro-Average		
	Precision	Recall	F-measure
RTE6 SVMs	70.32	32.59	44.54
RTE7 SVMs	68.01	28.29	39.96
RTE6 CRFs	56.04	38.31	45.51
RTE7 CRFs	45.14	43.27	44.18

Table 3: Performance (F1) of our system on RTE challenges, compared to other systems participated in these challenges. Median and Best indicate the median score and the highest score of all submissions, respectively

RTE challenge	Median	Best	Our system (SVMs)	Our system (CRFs)
RTE-6	33.72	48.01	44.54	45.51
RTE-7	39.89	48.00	39.96	44.18

We use the development set of RTE6, and develop set of RTE7, to train appropriate thresholds for the unseen test set of RTE6, and develop set of RTE7 separately. The micro averaged scores are shown in Table 2.

Our system's performance on the last two RTE challenges [1, 8] are presented in Table 3. Our results are better than the median of all submitted results.

Analyzing text and hypothesis pairs from the development set, we find that the negative pairs have very low word overlap, and there are fewer negative pairs with high overlap than positive pairs with high overlap, and there are more positive pairs with low overlap than positive pairs with high overlap.

Sometimes the *TH* pair's entailment relationship cannot be judged from surface features. The following *TH* pair shows system needs background knowledge to reason.

T: Kashmir is divided between Air and Islamic but both claim the region in its entirety.

H: Islamic and Air are rivals.

Islamic and Air both claim the region in its entirety, which means that the two countries has territorial disputes. According to human knowledge, we can judge that Islamic and Air are rivals.

If the *TH* pair involves synonyms conversion, the system mostly gives the wrong answer. The two countries as hostile can be introduced based on the following two sentences because rivals can be introduced by the intractable conflict.

T: The peace process has also led to resumption of trains, buses and flights between Air and Islamic, but the launching of the bus link across Kashmir _ severed in 1948 _ is more resonant, holding out some hope of an end to the intractable conflict over the region, which is claimed by both countries in its entirety.

H: Islamic and Air are rivals.

When there are many different forms for central words, the system often gives the wrong answer.

T: The two are also charged with conspiring in an explosion that killed two baggage handlers at Narita airport outside Tokyo 54 minutes before Flight 182 went down.

H: A bomb exploded at Narita airport.

In the above *TH* pair, explosion and exploded are the center words which have the same root, and then explosion and exploded should be matched.

The following *TH* pair involving center words, but Indian is not in *T*. So that *H* cannot be entailed by *T*.

T: Prime Minister Manmohan Singh's visit to Kashmir in April will be his third since he assumed office in May.

H: Manmohan Singh is the Indian Prime Minister.

5 Conclusions

This paper proposes a method to calculate the similarity between text *T* and hypothesis *H* based on the TF-IDF values. And then filters the sentence pairs that have high similarity score and are actually not entailed using machine learning methods. The experimental results show that our method is effective for recognizing textual entailment task. Our experiments also show that the different techniques that we employ perform very differently on some of the subsets of the recognizing textual entailment corpus and as a result, it is useful to use the nature of the dataset as a feature.

Acknowledgments

Part of this work was performed while the author was visiting USC/ISI, where the project was supported by the DARPA Machine Reading Program under AFRL prime contract no. FA8750-09-C-0172.

References

[1] Bentivogli L., Clark P., Dagan I. et al., The seventh pascal recognizing textual entailment challenge, in: Proceedings of TAC, 2011

[2] Houping Jia, Xiaojiang Huang, Tengfei Ma et al., PKUTM participation at TAC 2010 RTE and summarization track, in: Proceedings of TAC, 2010

[3] Moldovan D., Rus V., Logic form transformation of wordnet and its applicability to question answering, Annual meetting-association for computational linguistics, 39 (2001), 394–401

[4] Girju R., Automatic detection of causal relations for question answering, in: Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering, Association for Computational Linguistics, 12 (2003), 76–83

[5] Harabagiu S., Hickl A. Methods for using textual entailment in open-domain question answering, Annual meetting-association for computational linguistics, 44 (2006), 905

[6] Clinchant S., Goutte C., Gaussier E., Lexical entailment for information retrieval, Advances in Information Retrieval, Springer, 2006, 217–228

[7] Dagan I., Glickman O., Magnini B., The pascal recognising textual entailment challenge, Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment, Springer, 2006, 177–190

[8] Bentivogli L., Clark P., Dagan I. et al., The sixth pascal recognizing textual entailment challenge, in: Proceedings of TAC, 2010

[9] Cortes C., Vapnik V., Support-vector networks, Machine Learning, Springer, 20 (1995), 273–297

[10] Vapnik V., The nature of statistical learning theory, springer, 1999

[11] Lafferty J., McCallum A., Pereira F.C.N., Conditional random fields: probabilistic models for segmenting and labeling sequence data, 2001

[12] McCallum A., Freitag D., Pereira F., Maximum entropy Markov models for information extraction and segmentation, in: Proceedings of the Seventeenth International Conference on Machine Learning, 951 (2000), 591–598

[13] Chang C. C., Lin C. J., LIBSVM: a library for support vector machines, in: ACM Transactions on Intelligent Systems and Technology (TIST), ACM, 2 (2011), 27

[14] McCallum, A. K., Mallet: a machine learning for language toolkit, 2002