SJTU_CIT at TAC 2010: Guided Summarization Task

Peng Li and **Xinhua Zhang** and **Yinglin Wang**
Department of Computer Science and Engineering, Shanghai Jiao Tong University
{lipeng, atou, ylwang@sjtu.edu.cn}

Abstract

In this paper, we propose a approach to automatic generation of aspect-oriented summary from given newswire articles. We first develop an event-aspect LDA model to simultaneously cluster both sentences and words into aspects. We then combine them in Integer Linear Programming Framework for sentence selection. Key features of our method include automatic grouping of semantically related sentences and sentence compression on dependency tree. Although quantitative evaluation shows our method below average level, we still believe that this is one way towards to solve this problem.

1 Introduction

TAC2010 guided summarization task is to write a 100 word summary of a set of 10 newswire articles for a given topic, where the topic falls into a predefined category. A summary should cover all the aspects relevant to its category. Referring to accidents and natural disasters category, 7 aspects should be covered by the automatically generated summary. These aspects are what happened; date; location; reasons for accident/disaster; casualties; damages; rescue efforts/countermeasures. Additionally, an "update" component of the guided summarization task is to write a 100-word "update" summary of a subsequent 10 newswire articles for the topic, under the assumption that the user has already read the earlier articles.

1.1 Overview of Our Method

Our method focus on finding sentences which can summarize aspect content. We first develop an event-aspect LDA model to simultaneously cluster both sentences and words into aspects. Some aspects link to name entities like time or data, physical location, individual or groups participated in this event. Other aspects corresponding to relative long clause. We assume that one sentence contains one primary aspect and this aspect can be represent by several aspect words. Our summarization system follow extraction based procedure which consists of four main steps.

1. ***Sentence annotation***: We tag noun phrases with their ontological concepts, These concepts are Person, Location, Organization, and Data/Time. Also we tag other words as Aspect word, Document word or Background word using event-aspect LDA model, simultaneously this model can cluster sentences based annotated aspects (Li et al., 2010).

2. ***Sentence ranking***: We combine all above semantic evidences together for ranking.

3. ***Sentence compression***: We prune sentence on dependency tree instead of on parser tree, using grammatical relations to recognize clauses and remove redundant subtree from original dependency tree.

4. ***Sentence selection***: We select one compressed sentence for each aspect from clusters. Using Integer Linear Programming framework which optimize global objection function for sentence selection (McDonald, 2007; Gillick and Favre, 2009; Sauper and Barzilay, 2009).

2 Main Approach

2.1 Event-Aspect Model

We now formally present our event-aspect model. First, we assume that stop words can be identified using a standard stop word list.

We then assume that for a given topic category there are three kinds of unigram language models (i.e. multinomial word distributions). There is a background model $\phi^{\mathcal{B}}$ that generates words commonly used in all documents and all aspects. There are D document models ψ^d ($1 \leq d \leq D$), where D is the number of documents in the given summary collection, and there are A aspect models ϕ^a ($1 \leq a \leq A$), where A is the number of aspects. We assume that these word distributions have a uniform Dirichlet prior with parameter β.

Since not all aspects are discussed equally frequently, we assume that there is a global aspect distribution θ that controls how often each aspect occurs in the collection. θ is sampled from another Dirichlet prior with parameter α. There is also a multinomial distribution π that controls in each sentence how often we encounter a background word, a document word, or an aspect word. π has a Dirichlet prior with parameter γ.

Let S_d denote the number of sentences in document d, $N_{d,s}$ denote the number of words (after stop word removal) in sentence s of document d, and $w_{d,s,n}$ denote the n'th word in this sentence. We introduce hidden variables $z_{d,s}$ for each sentence to indicate the aspect a sentence belongs to. We also introduce hidden variables $y_{d,s,n}$ for each word to indicate whether a word is generated from the background model, the document model, or the aspect model. Figure 1 shows the process of generating the whole document collection. The plate notation of the model is shown in Figure 2. Note that the values of α, β and γ are fixed. The number of aspects A is also manually set.

2.2 Inference

Given a summary collection, i.e. the set of all $w_{d,s,n}$, our goal is to find the most likely assignment of $z_{d,s}$ and $y_{d,s,n}$, that is, the assignment that maximizes $p(\boldsymbol{z}, \boldsymbol{y}|\boldsymbol{w}; \alpha, \beta, \gamma)$, where $\boldsymbol{z}$, $\boldsymbol{y}$ and $\boldsymbol{w}$ represent the set of all z, y and w variables, respectively. With the assignment, sentences are naturally clustered into aspects, and words are labeled as either a background word, a document word, or an aspect word.

We approximate $p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{w}; \alpha, \beta, \gamma)$ by $p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{w}; \hat{\phi}^{\mathcal{B}}, \{\hat{\psi}^d\}_{d=1}^D, \{\hat{\phi}^a\}_{a=1}^A, \hat{\theta}, \hat{\pi})$, where

1. Draw $\theta \sim \mathrm{Dir}(\alpha), \phi^{\mathcal{B}} \sim \mathrm{Dir}(\beta), \pi \sim \mathrm{Dir}(\gamma)$
2. For each aspect $a = 1, \ldots, A$,
   (a) draw $\phi^a \sim \mathrm{Dir}(\beta)$
3. For each document $d = 1, \ldots, D$,
   (a) draw $\psi^d \sim \mathrm{Dir}(\beta)$
   (b) for each sentence $s = 1, \ldots, S_d$
      i. draw $z_{d,s} \sim \mathrm{Multi}(\theta)$
      ii. for each word $n = 1, \ldots, N_{d,s}$
         A. draw $y_{d,s,n} \sim \mathrm{Multi}(\pi)$
         B. draw $w_{d,s,n} \sim \mathrm{Multi}(\phi^{\mathcal{B}})$ if $y_{d,s,n} = 1$, $w_{d,s,n} \sim \mathrm{Multi}(\psi^d)$ if $y_{d,s,n} = 2$, or $w_{d,s,n} \sim \mathrm{Multi}(\phi^{z_{d,s}})$ if $y_{d,s,n} = 3$

Figure 1: The document generation process.

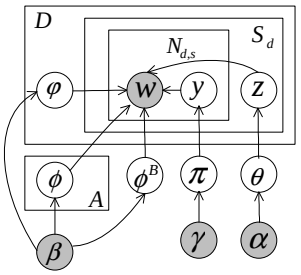

Figure 2: The entity-aspect model.

$\hat{\phi}^{\mathcal{B}}$, $\{\hat{\psi}^d\}_{d=1}^D$, $\{\hat{\phi}^a\}_{a=1}^A$, $\hat{\theta}$ and $\hat{\pi}$ are estimated using Gibbs sampling, which is commonly used for inference for LDA models (Griffiths and Steyvers, 2004). Due to space limit, we give the formulas for the Gibbs sampler below without derivation.

First, given sentence s in document d, we sample a value for $z_{d,s}$ given the values of all other z and y variables using the following formula:

$$p(z_{d,s} = a|\boldsymbol{z}_{\neg\{d,s\}}, \boldsymbol{y}, \boldsymbol{w}) \propto \frac{C^{\mathcal{A}}_{(a)} + \alpha}{C^{\mathcal{A}}_{(\cdot)} + A\alpha} \cdot \frac{\prod_{v=1}^{V} \prod_{i=0}^{E_{(v)}} (C^a_{(v)} + i + \beta)}{\prod_{i=0}^{E_{(\cdot)}} (C^a_{(\cdot)} + i + V\beta)}.$$

In the formula above, $\boldsymbol{z}_{\neg\{d,s\}}$ is the current aspect assignment of all sentences excluding the current sentence. $C^{\mathcal{A}}_{(a)}$ is the number of sentences assigned to aspect a, and $C^{\mathcal{A}}_{(\cdot)}$ is the total number of sentences. V is the vocabulary size. $C^a_{(v)}$ is the number of times word v has been assigned to aspect a. $C^a_{(\cdot)}$ is the total number of words assigned to aspect a. All the counts above exclude the current sentence. $E_{(v)}$ is the number of times

word v occurs in the current sentence and is assigned to be an aspect word, as indicated by $\boldsymbol{y}$, and $E_{(\cdot)}$ is the total number of words in the current sentence that are assigned to be an aspect word.

We then sample a value for $y_{d,s,n}$ for each word in the current sentence using the following formulas:

$$p(y_{d,s,n}=1|\boldsymbol{z},\boldsymbol{y}_{\neg\{d,s,n\}}) \propto \frac{C^{\pi}_{(1)}+\gamma}{C^{\pi}_{(\cdot)}+3\gamma}\cdot\frac{C^{\mathcal{B}}_{(w_{d,s,n})}+\beta}{C^{\mathcal{B}}_{(\cdot)}+V\beta},$$

$$p(y_{d,s,n}=2|\boldsymbol{z},\boldsymbol{y}_{\neg\{d,s,n\}}) \propto \frac{C^{\pi}_{(2)}+\gamma}{C^{\pi}_{(\cdot)}+3\gamma}\cdot\frac{C^{d}_{(w_{d,s,n})}+\beta}{C^{d}_{(\cdot)}+V\beta},$$

$$p(y_{d,s,n}=3|\boldsymbol{z},\boldsymbol{y}_{\neg\{d,s,n\}}) \propto \frac{C^{\pi}_{(3)}+\gamma}{C^{\pi}_{(\cdot)}+3\gamma}\cdot\frac{C^{a}_{(w_{d,s,n})}+\beta}{C^{a}_{(\cdot)}+V\beta}.$$

In the formulas above, $\boldsymbol{y}_{\neg\{d,s,n\}}$ is the set of all y variables excluding $y_{d,s,n}$. $C^{\pi}_{(1)}$, $C^{\pi}_{(2)}$ and $C^{\pi}_{(3)}$ are the numbers of words assigned to be a background word, a document word, or an aspect word, respectively, and $C^{\pi}_{(\cdot)}$ is the total number of words. $C^{\mathcal{B}}$ and C^{d} are counters similar to C^{a} but are for the background model and the document models. In all these counts, the current word is excluded.

With one Gibbs sample, we can make the following estimation:

$$\hat{\phi}^{B}_{v}=\frac{C^{\mathcal{B}}_{(v)}+\beta}{C^{\mathcal{B}}_{(\cdot)}+V\beta}, \hat{\psi}^{d}_{v}=\frac{C^{d}_{(v)}+\beta}{C^{d}_{(\cdot)}+V\beta}, \hat{\phi}^{a}_{v}=\frac{C^{a}_{(v)}+\beta}{C^{a}_{(\cdot)}+V\beta},$$

$$\hat{\theta}_{a}=\frac{C^{\mathcal{A}}_{(a)}+\alpha}{C^{\mathcal{A}}_{(\cdot)}+A\alpha}, \hat{\pi}_{t}=\frac{C^{\pi}_{(t)}+\gamma}{C^{\pi}_{(\cdot)}+3\gamma}(1\leq t\leq 3).$$

Here the counts include all sentences and all words.

In our experiments, we set $\alpha = 5$, $\beta = 0.01$ and $\gamma = 20$. We run 100 burn-in iterations through all documents in a collection to stabilize the distribution of $\boldsymbol{z}$ and $\boldsymbol{y}$ before collecting samples. We take 10 samples with a gap of 10 iterations between two samples, and average over these 10 samples to get the estimation for the parameters.

After estimating $\hat{\phi}^{\mathcal{B}}$, $\{\hat{\psi}^{d}\}^{D}_{d=1}$, $\{\hat{\phi}^{a}\}^{A}_{a=1}$, $\hat{\theta}$ and $\hat{\pi}$, we find the values of each $z_{d,s}$ and $y_{d,s,n}$ that maximize $p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{w};\hat{\phi}^{\mathcal{B}},\{\hat{\psi}^{d}\}^{D}_{d=1},\{\hat{\phi}^{a}\}^{A}_{a=1},\hat{\theta},\hat{\pi})$. This assignment, together with the standard stop word list we use, gives us sentences clustered into A aspects, where each word is labeled as either a stop word, a background word, a document word or an aspect word.

2.3 Sentence Annotation

Firstly of all, we remove sentence which its length is less then 5 words before tagging sentence for each category. Then Tagging words as Aspect word, Document word or Background word using our event-aspect LDA model. After that, we map document words and background words to their ontological concept like person, location, organization or time. To tag location and organization, we use Stanford NER tagger, we use heuristic rules coded by perl scripts to tag time, and make use of WordNet Hypernym relation to tag person.

2.4 Sentence Ranking

For each sentence grouped by our Event-aspect LDA model, we want to get most representative sentence based on annotation information. For each aspect A_j , we set the score of Sentence S_{jl} using below formula,

$$Score(S_{jl}) = \sum_{\alpha,\beta\in S_{jl}} \omega_{\alpha} + \nu_{\beta}$$

where ω_{α} is the probability of aspect word which is estimated by our LDA model, ν_{β} is the probability of concept word which is estimated by below formula,

$$\nu_{\beta} = \frac{n_{\beta}}{\sum n_{\beta}}$$

where n_{β} is the number of concepts β in sentence S_{jl}, $\sum n_{\beta}$ is total numbers concepts in each topic.

2.5 Sentence Compression

Instead of recognizing clauses from parser tree (Zajic et al., 2007), we extract clauses from dependency tree. For instance, We have friends whose children go to Columbine , the freshman said .. We want to remove the freshman said, if the procedure run on parser tree of the sentence, it may get whose children go to Columbine using SBAR label to match, so we leverage English grammatical rules to find clause, the procedure is below,

1. Select possible subtree root nodes using "ccomp", "parataxis" or "complm" grammatical relations which are defined by Stanford typed dependencies manual.

2. Decide which subtree root could be clause's root, if this root contain maximum number of children nodes and the collection of all children edges include "obj" or "aux" relations, it is selected as clause's root.

3. Travel the subtree to extract clause from original sentence.

2.6 Sentence Selection

We choose sentence which cover more relevant aspect words as well as this sentence should get higher ranking score in the cluster. We model this selection process under Integer Linear Programming framework. For each aspect A_j in particular category, we have aspect word α_i represent the indicator of specific aspect word, and S_{jl} is the indicator of specific sentence, l is the ranked position in this cluster. Below are the objective function and constraints,

$$\max(\sum_{i=1}^{n} \omega_{\alpha_i} \cdot \alpha_i - \sum_{j=1}^{K}\sum_{l=1}^{R} l \cdot S_{jl}) \quad (1)$$

$$\sum_{j=1}^{K}\sum_{l=1}^{R} leg_{jl} \cdot S_{jl} \leq L \quad (2)$$

where leg_{jl} is the length of S_{jl}

$$\sum_{l=1}^{R} S_{jl} = 1 \qquad \forall j \in 1 \ldots K \quad (3)$$

$$S_{jl} \cdot OCC_{jil} \leq \alpha_i \qquad \forall i, l \quad (4)$$

$$\sum_{l=1}^{R} S_{jl} \cdot OCC_{jil} \geq \alpha_i \qquad \forall i \quad (5)$$

where OCC_{jil} is the occurrence of α_i in sentence S_{jl}

2.7 Update Summary

To solve updated summarization task, we combine Set A and Set B together to do labeling under the assumption that the user want to summarized Set B after he already Set A. We then separate Set B labeling results from Set A, later procedures are the same as guided-summarization task.

	average ROUGE-2 recall		average ROUGE-SU recall	
	A	B	A	B
Run-1	0.04411	0.02845	0.07762	0.06690
Run-2	0.04055	0.02680	0.07589	0.06424

Table 2: Rouge Results

	average BE recall	
	A	B
Run-1	0.02316	0.01251
Run-2	0.02473	0.00949

Table 3: BE Results

3 Evaluation

TAC 2010 provides 46 topics for evaluation. Each topic includes a topic statement and 20 relevant documents which have been divided into 2 sets: Document Set A and Document Set B. Each document set has 10 documents, and all the documents in Set A chronologically precede the documents in Set B. Eight NIST assessors selected and wrote summaries for the 46 topics in the TAC 2010 guided summarization task, and assessors wrote 4 model summaries for each docset. NIST received 41 runs from 23 participants for the guided summarization task. NIST evaluated all summaries manually for overall responsiveness and for content according to the Pyramid method. All summaries were also automatically evaluated using ROUGE/BE.

In Run-1(summarizer ID is 39), we set the number of average aspect cluster for each category is 5 inside our Event-aspect LDA Model, we set this value is 4 in Run-2(summarizer ID is 7). Table 1 is manual evaluation results, Table 2 is the ROUGE results, Table 3 is the BE results.

	average modified (pyramid) score		average numSCUs		average numrepe-titions		macroaverage modified score with 3 models		average linguistic quality		average overall respon-siveness	
	A	B	A	B	A	B	A	B	A	B	A	B
Run-1	0.187	0.085	2.804	1.196	0.521	0.043	0.184	0.083	2.435	2.435	1.935	1.543
Run-2	0.154	0.075	2.304	1.000	0.087	0.022	0.151	0.073	2.043	2.130	1.739	1.413

Table 1: Manual Evaluation

3.1 Analysis

The official evaluation results presented in the above tables show that our system gets lower performance which is below the average. One reason is that precisely labeling need more documents, these aspects can't recognized easily by our LDA model, so many wrong concepts are labeled. This lead to our sentence selection model may not work very well.

4 Conclusions

In this paper, we develop an Event-aspect LDA model to automatically generate aspect-oriented summary. We model this aspect-oriented sentence selection process under Integer Linear Programming framework. We also propose a approach to do sentence compression using dependency parser tree. However, the evaluation results is bad, modeling the aspects of summary need combine more domain knowledge, we need develop a new event-aspect model which can using domain knowledge.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC No. 60773088), the National High-tech R&D Program of China (863 Program No. 2009AA04Z106), and the Key Program of Basic Research of Shanghai Municipal S&T Commission (No. 08JC1411700).

References

Gillick, Dan and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 10–18.

Griffiths, Thomas L. and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl. 1):5228–5235.

Li, Peng, Jing Jiang, and Yinglin Wang. 2010. Generating templates of entity summaries with an entity-aspect model and pattern mining. In *Proceedings of the Joint Conference of the 48th Annual Meeting of the ACL*. Association for Computational Linguistics.

McDonald, Ryan. 2007. A study of global inference algorithms in multi-document summarization. *Advances in Information Retrieval*, pages 557–564.

Sauper, Christina and Regina Barzilay. 2009. Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216, Suntec, Singapore, August. Association for Computational Linguistics.

Zajic, D., B.J. Dorr, J. Lin, and R. Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing & Management*, 43(6):1549–1570.

SJTU_CIT at TAC 2010 RTE Track

Xinhua Zhang, Peng Li and Yinglin Wang
Dept. of Computer Science and Engineering
Shanghai Jiao-Tong University
No. 800, Dongchuan Road, Shanghai, China 200240
{atou, lipeng, ylwang}@sjtu.edu.cn

***Abstract*—in this paper we focus on finding a best semantic alignment between two sentences. Because alignments are too many to enumerate, the alignment method must be effective. Based on this alignment we integrated several methods to support and perform recognizing textural entailment (RTE). Finally, we implement our method on to RTE6 Test Data. The result shows that word alignment based RTE has a good performance.**

I. INTRODUCTION

Recognizing textural entailment (RTE) has been paid more and more attention in recent years. The definition of this task is that given two sentences, a RTE system works out a judgment that whether one sentence can be inferred from the other. For examples,

T: He bought a pen in the store.

H: He owes a pen.

Obviously, T infers H. This task is difficult because it is more than comparing the word similarity between two sentences. It needs in-depth inference, for instance, from *owe* happens after *buy* we know that these two sentences has a chronological relation. RTE is useful in many Natural Language Processing applications, such as Question Answering, Information Retrieval, Information Extraction, and Text Summarization.

In RTE, we believe that there are three major problems needed to be solved. They are word alignment, word similarity problem or word relation problem, and inference. We compare them to human judging process. When a human decides whether a sentence can infer others, he should compare the same and similar places in two sentences first. This process can be viewed as word alignment. Then, the relations between two words or phrases help him make a decision. Finally, he constructs concept for each sentence and compares them with the notion of alignment.

In this paper, we focus on the first problem that is finding a best alignment. However, finding a right match is difficult. The possible matches could be exponential in the number of words. To solve it effectively is the key to this problem. Hence, *Structured Prediction* [2] is introduced to relieve this issue. It claims that weighted matching, such as word alignment, is able to be converted into optimization problems and solved in polynomial time. In [1], it proposes a svm-like classifier. It uses maximize margin method to learn a structure model. With this learned model, the structure svm can output a best structure, or a matching.

This paper also proposes an integrated method to handle RTE problem, involves word alignment and word similarity calculation based on several knowledge bases. (System architecture is illustrated in Fig.1) Words similarity is very important in RTE. In consideration of words with different POS or phrase, we combine different knowledge bases. Currently there is no such a knowledge base that can provide enough information in calculation similarity. Thus for different situation, we use different knowledge base.

The alignment method is regardless of syntax information. Therefore, it is inaccurate sometimes. And for those sentences that are not suited for word alignment, the result might be undesirable. In order to cope with this problem, we add some simple syntax-based rule and combine it with the alignment result to make the final decision.

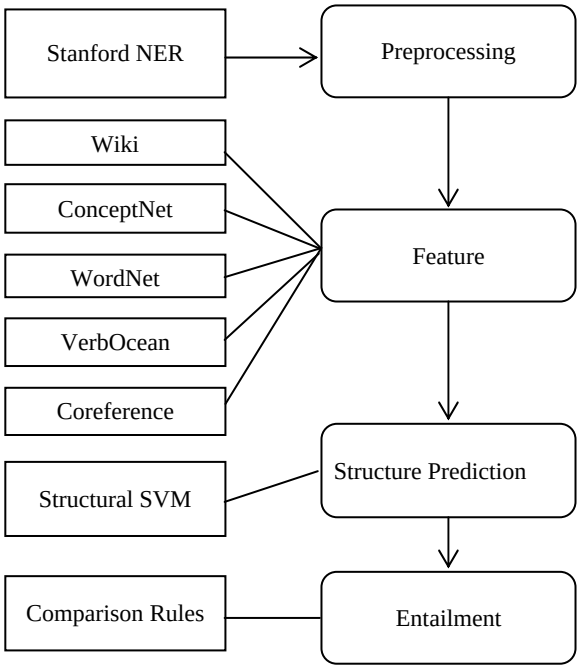

Figure 1. System Frame Work

In RTE6, the data is extracted from an article. Sometimes the information of the sentence is incomplete without referencing to other sentences in the article. In our system we use BART [15], a coreference tool that can parse pronouns and other coreference.

II. Related Work

The development of RTE is mostly motivated by PASCAL challenge [12]. Many methods had been proposed, and they can be roughly divided into two types.

A. Rule based matching

In RTE rule based matching focus primarily on how to find a feasible matching, the final entailment decision can be made easily through the summation of each matching's weight. In [7], they construct two sentences' dependency trees and exclude those with low relatedness pair based on a tree edit distance algorithm. After that they match every node in dependency trees associated with the two sentences. Beside dependency tree, [9] defines predicate-argument graph (PAG) that decomposes the two sentences into two sets of graphs. The overlap rate of elements in these two sets represents the relatedness between two sentences. In [6] they favor shallow semantic structure, with this Semantic Role Labeling based structure it is not difficult to match between the same semantic roles.

B. Feature based classification

In RTE problem, the output of the system is just simple yes or no. This characteristic shows that it can be solved via machine learning approach, especially classifiers such as SVM. In [5], they define sentence-level features and path-level features. The former is primarily lexical similarity, while the latter is concerned syntax similarity.

Many works had been put in structure prediction [1] [2]. These methods solve the problem effectively, within polynomial time.

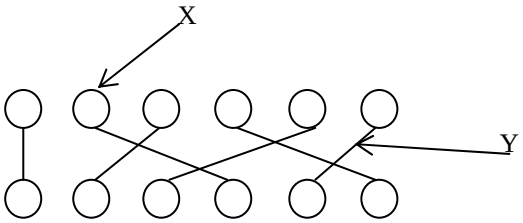

Figure 2. Structural Model

III. Word Alignment

We use the Struct-SVM[1] as the main tool for finding weights for similarity features. It is a discriminative method. Suppose we have a structure pairs (x,y), where x can be a set of nodes, y is a set of labels which of each connects two nodes in x. (Fig. 2) Then we define a score function

$$\text{score}(x, y) = \omega^{T}\Phi(x, y)\epsilon R \tag{1}$$

Where $\Phi(x,y)\epsilon R^{n}$ is a function that convert the relation between x and y into a feature vector, ω is a vector of weights for each feature and n is feature numbers. It gives us a score of a y applying to x. Our objective is to find the best structure.

$$y^{*} = \text{argmax}_{y\epsilon Y}\, \omega^{T}\, \Phi(x, y) \tag{2}$$

Where Y is the output space, generally it is very large. For some kinds of structure, (2) can be viewed as a certain optimization problem. The optimizing process is also called "prediction".

The Struct-SVM is just a frame to learn the weight vector ω for different features and calculates y^{*}. We need an effective method to find the best alignment through enormous ones. For simplicity, it is only considered as the weighted bipartite matching problem [2]. We let $z_{ij} = 1$ represent one of the labels in y that connect nodes i and j in sentences pair in x, whereas $z_{ij} = 0$ means there is no connection. Thus we have a combinational optimization problem below.

$$\max_{i,j} c_{ij} z_{ij} \tag{6}$$

$$\text{s.t.} \sum_{i=1}^{n} z_{ij} \quad j = 1, \dots, n \quad \sum_{j=1}^{n} z_{ij} \quad i = 1, \dots, n$$

Where c_{ij} is the weight for label z_{ij}. This problem can be solved by using Hungarian Method [13].

IV. Alignment Features

To predict alignment structure in sentences pairs, we need sets of features in the mode. Word similarity based on WordNet has been widely used in RTE. Its strength is that it contains a large amount of words, by calculating the distance between two words it's easily to get the similarity.

However, in alignment in RTE is usually between words that have different POS or between single word and phrase. For example, *Hunter* has some relation to phrase *Killing a prey*. Therefore, we introduce a knowledge base called ConceptNet. As we can see in Fig. 3 that ConceptNet provides us with the attributes of an object or an action.

Figure 3. ConceptNet Examples

Although WordNet includes many words, yet in RTE we may have to find the relation between Norn phrase and because Wikipedia is one of the largest electronic encyclopedias, its content is mainly consist of Noun-related knowledge. [4] computes two phrase relation by representing them with two vector of wiki entry and calculating the similarity between them.

Verb and verb similarity is also important in decision making. The knowledge bases listed above are relatively week in comparing verbs. Thus a few works had put effort in extracting verb pair through plain text. Here we use

VerbOcean[9], it is a broad-coverage semantic network of verbs, it defines several relations between verb.

By observing the RTE6 data we can easily find that it's hard to entail without referencing to other sentences in the same article. We simply define the coreference feature as that if two phrases are coreferenced decided by BART or if they appear in the same sentence in the article, we believe they have certain relation and they are coreferenced.

In addition, we use a feature to represent that two stemmed words are identical.

V. Comparison Rules

The natural language's pattern varies greatly in different situations. In our system, we only define several rules to apply to some common language pattern.

A. Set Comparison Rule

Descripting a set is quite common in our communication. This process usually starts with certain key words i.e. include, contain, or punctuation i.e. ':'. By identifying them we know there is a set, and we can perform comparison according to set theory.

B. Subject and Object

We find out similar verbs based on the alignment, and then compare their subjects and objects. For the situation that a verb is aligned to a noun, we believe that the noun is a mention of an event and by looking at the relation between the verb's subject or object and the event, we can make a judgment.

VI. RTE System

A. Preprocessing

Usually texts for entailment are with different length. However, the Hungarian method that used to find the best alignment is able to solve bipartite weighted matching that has the same length. Therefore we add some blank nodes. Every node that connects to these blank nodes has zero weight.

Bipartite matching has a disadvantage, that is when it encounter a Norn phrase or Proper Norn that are more than one word, it is not meaningful to align only one of the word. We use the Stanford Name Entity Tagger [14], to extract those name entity, Norn phrase and treat them as just one node in matching.

B. Entailnment

After getting all features and alignment, we sum up all edges' weight and divide the result by the length of the shortest sentences, because long sentences tend to have lager sum of weights, and we set two thresholds according to experience. Candidates with scores higher than the higher threshold will be marked as TRUE; and candidates with scores lower than the lower threshold will be marked as FALSE. This process can rule out candidates that have high possibility for entailment and those irrelevant ones. Then we combine rules defined above and alignment result to perform a deeper inference.

VII. Submissions and Result

A. Main Test

We submitted three runs, executed with different thresholds. The results are shown in TABLE I. We can see that we still have many works to do to improve the performance of our system. Compared to past RTE result, these results are relatively poor. We believe that it is mainly due to the sentences' incomplete information. And our coreference feature can only handle a small amount of situations.

TABLE I. RTE6 Testing Set Result

RUN ID	Micro Average Precision	Recall	F-measure
RUN1	0.2634	0.5778	0.3618
RUN2	0.3209	0.4995	0.3907
RUN3	0.3436	0.4667	0.3957
F_{high}	0.4801		
F_{Median}	0.3372		
F_{Low}	0.1160		

B. Ablation Test

In this test we submitted three runs with different knowledge resources as shown in TABLE II. They all have negative impact of Micro Average Precision, but positive impact on Recall. With this result we can infer that the structure that our system produces is mainly decided by words or phrases that are identical. And these resources are able to help the system to find the semantic relation between two sentences, however, in lexical level.

TABLE II. Ablation Test Result

Resource	Micro Average Precision	Recall	F-measure
WordNet	-0.0663	0.0847	0.0003
Wikipedia	-0.0514	0.1545	0.0470
VerbOcean	-0.0272	0.0148	-0.0115

VIII. Conclusion and Future Work

This paper proposes a frame work for RTE which depends on word alignment. By applying a discriminative method we can solve the bipartite matching problem efficiently. On it we apply rule based entailment. There are still many part of the system needed to be improved. Here we mainly use lexical features, which are only part of features in Natural Language Processing. Their defects are revealed by the ablation test. In the future, we will add some syntax features. And the Hungarian method could only solve bipartite matching problem. Although we use Stanford NER to extract name entities and put them into one node, yet some phrase still cannot be recognized. Also the coreference problem is an issue that has been put a lot of efforts, and the RTE system will rely heavily on it when facing phrases extracted from an article.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (NSFC No. 60773088), the National High-tech R&D Program of China (863 Program No. 2009AA04Z106), and the Key Program of Basic Research of Shanghai Municipal S&T Commission (No. 08JC1411700).

REFERENCES

[1] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support Vector Learning for Interdependent and Structured Output Spaces, ICML, 2004.

[2] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: a large margin approach. In Proceedings of the International Conference on Machine Learning. 2005.

[3] B. Taskar, S. Lacoste-Julien, D. Klein. A Discriminative Matching Approach to Word Alignment. Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 73–80, Vancouver, October 2005.

[4] Gabrilovich, E. and S. Markovitch.. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07), 2007.

[5] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediciton models: A large margin approach. In ICML 22, Bonn, Germany, 2005.

[6] F. LI et al. THU QUANTA at TAC 2009 KBP and RTE Track. In Proceedings of Text Analysis Conference 2009.

[7] M Sammons et al. Relation Alignment for Textual Entailment Recognition. In Proceedings of Text Analysis Conference 2009.

[8] Iftene, A. Uaic participation at rte5. In Proceedings of Text Analysis Conference 2009.

[9] T. Chklovski and P. Pantel. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. Proc. EMNLP-2004.

[10] Rui Wang and Yi Zhang. 2009. Recognizing textual relatedness with predicate-argument structures. In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2009), Singapore, Singapore. Association for Computational Linguistics.

[11] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. In Qui˜nonero-Candela et al., editor, MLCW 2005, LNAI Volume 3944, pages 177–190.Springer-Verlag.

[12] http://pascallin.ecs.soton.ac.uk/Challenges/RTE/

[13] A. Schrijver. Combinatorial Optimization: Polyhedra and Efficiency. Springer, 2003.

[14] http://nlp.stanford.edu/ner/index.shtml

[15] http://bart-anaphora.org/