PRIS at TAC 2009: Experiments in KBP Track

Si Li, Sanyuan Gao, Zongyu Zhang, Xinsheng Li, Jingyi Guan, Weiran Xu, Jun Guo
Pattern Recognition and Intelligent System Lab
Beijing University of Posts and Telecommunications
Beijing, P.R. China, 100876
E-mail: ls198cf@gmail.com, sanyuanv@163.com

Abstract:

This paper describes BUPT (pris) participation in entity linking task and slot filling task. The system adopts a two-stage strategy in entity linking task and slot filling task. In the first stage, the system carries out a basic topic relevance retrieval to get top k documents for each query. In the second stage, cross-document coreference resolution is based on automatic text summary and automatic entity relation extraction is based on CRFs.

1 Introduction

The KBP track had two tasks in the TAC 2009. We participated in both entity linking task and slot filling task [1]. The entity linking task is to determine for each target list entry, which entity is being referred to. We considered this task as a cross-document coreference resolution task. The slot filling task involves learning a pre-defined set of relationships and attributes for target entities based on the evaluation corpus. This task is regarded as a relation extraction task about a given target entity.

The PRIS system submitted by PRIS lab at Beijing University of Posts and Telecommunications adopts a two-stage strategy. In the first stage, the basic ad-hoc retrieval platform is based on the Indri Retrieval Toolkit [2]. The system carries out a basic topic relevance retrieval to get the top 10 documents for each query. In the second stage, cross-document coreference resolution is based on automatic text summary for the entity linking task and automatic entity relation extraction is based on CRFs for the slot filling task. In entity linking task, we propose a cross-document coreference resolution algorithm based on automatic text summary instead of the original text. In our approach, we extract query-specific and informative-indicative summary from the original text by using Hobbs algorithm and measure the similarity between two summaries. In slot filling task, we consider this task as relation extraction task. Our system combines CRFs-based classifiers to implement relation extraction. These classifiers are trained mainly on the document set extracted from KB corpus according to the labels in each node.

The remainder of this paper is organized as follows. In section 2, the automatic text summary-based cross-document coreference resolution (ATSCDCR) system for entity linking task is presented. Section 3 describes the slot filling part. Evaluation results are shown in section 4.

2 Entity Linking

Fig. 1 shows the framework of ATSCDCR system. Our approach consists of four primary steps: Entity Retrieval, Entity Type Recognition, Summarization and Coreference Decision.

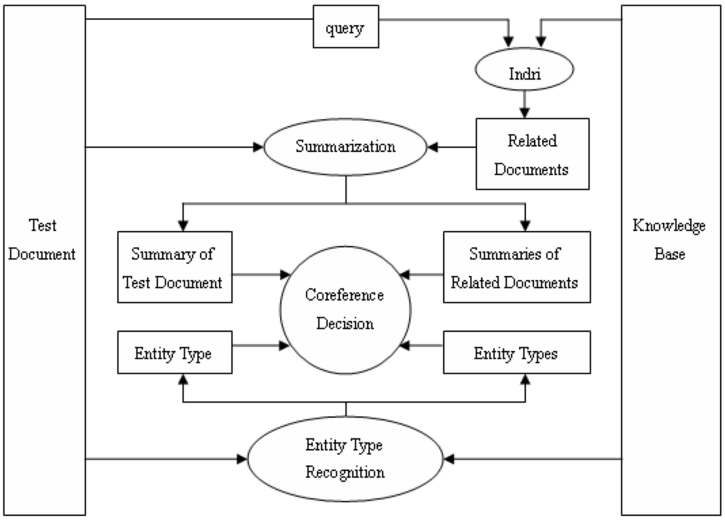

Figure 1: Framework of ATSCDCR System

2.1 Entity Retrieval

The knowledge base is always on the order of millions of entities. It is quite time-consuming to traverse all the underlying entities when resolving an entity mention. It is assumed that the authority file of target entity has at least one token in the mention name of test document, so we can retrieve finite entity candidates from the knowledge base with the mention name be the query. To this end, Indri is selected as our information retrieval platform, which is based on language model and Inference Network.

2.2 Named-Entity Type Recognition

The entity types in our experiment may be Person, Organization and Geo-Political. If the type of one target entity is uncertain, we regard it to be Unknown (UKN). In order to improve the accuracy of the resolution, entity type is identified by Stanford Named Entity Recognizer [3] before Coreference Decision.

2.3 Summarization

The documents in the test set often come from news, blog or forum; it is quite probable that one of them consists of several parts of different content. In light of this, we leverage the query-specific summary instead of the original text for similarity measure between two documents; different queries may produce different summaries of the same original text. Automatic Text summarization is an important part in ATSCDCR system; it employs intra-document coreference resolution technology, which now is an increasingly active research area.

2.3.1 Intra-Document Coreference Resolution

Intra-document coreference resolution is mapping intra-document mentions into the entities that they are referring to. It has important applications in areas such as question answering, machine translation and automatic summarization.

The dominant approach for intra-document coreference resolution is to decompose coreference resolution task into a collection of pairwise Coreference Decisions, and then apply discriminative learning methods to pairs of mentions. This kind of methods requires plentiful labeled data as well as substantial calculation, so an unsupervised method is used to resolve the pronominal anaphora in our work. Unsupervised methods are usually linguistically motivated methods; it leverages the syntactic information and semantic information of sentences.

2.3.2 Hobbs algorithm

Hobbs algorithm [4] is proposed by Hobbs in 1978 for the resolution of pronominal coreference in English. It is based on searching for a pronoun's anaphora in the syntactic parse tree of input sentences. Hobbs Naive algorithm used in ATSCDCR system makes use of syntactic information rather than semantic information. We parse the sentences using The Stanford Parser: A Statistical Parser [5].

Hobbs' Naïve algorithm is divided into nine steps. Step 1 finds the NP node immediately dominating the pronoun; Step 2 and Step 3 deal with the case when the anaphora and the candidate antecedent in the same level in a parse tree; Step 4 works in the situation where the antecedent and the pronominal anaphora are of different sentences. The other steps are not considered for automatic summarization, because we just focus on the relationship between two adjacent sentences. If one pronoun in current sentence is referring to an antecedent in previous sentence, true is returned or false is returned. The simplified Hobbs Naïve algorithm applied in ATSCDCR system is given in Fig. 2.

1. Begin at NP node immediately dominating the pronoun in the parse tree of S.
2. Go up the tree to the first NP or S node encountered. Call this node X and call the path to reach it p.
3. Traverse all branches below node X to the left of path p in left-to-right, breadth-first fashion. If an NP node is encountered with an NP or S node between it and X, then find next NP node, and continue to step 2.
4. If node X is not the highest S node in the sentence, then find next NP node, and go to step 2. Otherwise traverse the parse tree of previous sentence in a left-to-right, breadth-first manner, and when an NP node is encountered, return true.
5. Return false.

Figure 2. Simplified Hobbs Naïve algorithm

2.3.3 Summary Extraction

In ATSCDCR system, we concentrate on choosing good sentences for summary. The summary will be informative-indicative and query-specific. Details of the summary extraction algorithm are described as follows.

First, one sentence is the summary sentence if it contains at least one word of query.

Next, the summary sentence is one sentence with one pronoun of it referring to an antecedent in the previous summary sentence. The simplified Hobbs' Naïve algorithm introduced in 2.3.2 is used here for pronoun resolution.

Thirdly, one sentence is not summary sentence if it does not meet the two requirements above.

Sometimes, no summary sentence can be extracted by using our algorithm if there is no query term in the document. In such cases, the original text is used instead of the summary.

2.4 Coreference Decision

Instead of the original documents, the summaries are used for similarity measure in ATSCDCR system. Two algorithms are introduced for similarity calculation; they are the Vector Space Model and the KL divergence Model. A comparison between them is analyzed in the experiments. The summary in these two algorithms is used as the vector of terms, which have been preprocessed by removing stop words and stemming by Porter's Algorithm.

2.4.1 The Vector Space Model

We denote by $\vec{V}(S)$ the summary vector of document *D*, and the cosine similarity between two documents D_1 and D_2 is computed as:

$$Sim(S_1, S_2) = \frac{\vec{V}(S_1) \bullet \vec{V}(S_2)}{|\vec{V}(S_1)||\vec{V}(S_2)|} = \sum_{common\ terms:\ t_j} w_{1j} \times w_{2j}, \tag{1}$$

where t_j is a term present in both S_1 and S_2, w_{1j} is the weight of the term t_j in S_1 and w_{2j} is the weight of t_j in S_2. The weight of a term t_j in the vector $\vec{v}(S)$ is given by:

$$w_j = tf_j / \sqrt{\sum_{i=1}^{M} tf_i^{\,2}}, \tag{2}$$

where tf_i is the frequency of the term t_i in the summary.

2.4.2 The KL divergence Model

In probability theory and information theory, the Kullback-Leibler divergence is a non-symmetric measure of the difference between two probability distributions *P* and *Q*. Here we use improved KL divergence Model to measure the similarity between two documents. It is defined to be

$$D_{KL}(P \| Q) = \sum_i (P(i) - Q(i)) \log \frac{P(i)}{Q(i)}, \tag{3}$$

where *P* stands for the distribution of terms in the summary of document D_1, *Q* stands for the distribution of terms in the summary document D_2, word *i* occurs in D_1 or D_2.

Different from the KL divergence, the improved KL divergence formula is Symmetrical and Non-negative. The more close to 0 the value is, the more similar the two documents are.

2.4.3 Decision strategy

In ATSCDCR system, a Coreference Decision is made by combining the entity type recognition and similarity measure. Two entity mentions are coreferent to the same entity only in the case when they have high similarity measure and matched entity type. Matched entity type does not only mean the complete same type, but also refers to that one mention type is UKN.

3 Slot filling

This slot filling system consists of four primary steps: Entity Retrieval, Entity Type Recognition, Relation extraction and Resolution Decision. Entity Retrieval and Entity Type Recognition are the same as the ATSCDCR system. In this session, based on the Indri retrieval system, we get the top 10 documents for each target entity in test collection. Then, entity type is recognized in top documents. In the following, relation extraction is shown in detail.

3.1 Relation extraction

The unsupervised machine learning method and semi-supervised machine learning method are used for relation extraction.

For unsupervised machine learning, we design some trigger words to help to fill the slots. When the trigger word appears with entity which has suitable type, the slot can be filled by this entity. Take the trigger word “born” for an example, if the entity around “born” has the type of GPE, it can be put into the “birth place” slot and it will be put into the “birth date” slot if the entity has the type of Time.

In semi-supervised machine learning method, the feature-based methods transform the context into features. We regard the relation extraction problem as a classification learning problem. Features are selected as the following:

1. Named entity pairs: the target entity context and relation entity context, they are tagged as TE and RE respectively;
2. Previous word features: three words in front of TE or RE are selected as features;
3. Previous POS features: the POS tags of the previous word features;
4. Next word features: three words next to TE or RE are selected as features;
5. Next POS features: the POS tags of the next word features;
6. Sequence feature: the sequence between TE and RE;
7. Named entity pairs location feature: the position of the Named entity pairs in the sentence;
8. Other entity feature: if there is another named entity between the named entity pairs, the feature is set to 1;
9. Number feature: the word numbers between TE and RE;
10. Appearance feature: if the named entity pairs appear in the same sub-sentence, the feature is 1;
11. Type feature: the entity type of TE and RE, such as PER;
12. Verb feature: the verb context;
13. Verb location feature: the position of the verb word in the sentence;
14. Order feature: order of named entity and verb word;

With these features, we train the classifiers based on CRFs. Then we do the relation extraction in the top documents.

3.2 Resolution Decision

This step is used to select a better filling result between unsupervised machine learning method and semi-supervised machine learning method. We follow these measures:

1. If the slot can be filled by unsupervised machine learning method, this result is preferred.
2. If the slot can’t be filled by unsupervised machine learning method, we select the semi-supervised machine learning method result. To deal with the problem that the same slot has more than one different result, we design a function as the following:

$$S = \mu * Sim + (1-\mu) * P, \tag{4}$$

S is the score of the relation type which is learnt from the classification. *Sim* means the similarity between query and document, and the similarity score is got from Indri. *P* is the probability to be judged as this relation type by CRFs. μ is a weighting parameter distributing in the interval [0, 1]. μ is a parameter balancing the scores of similarity and probability. According to this score, the top 1 result is selected to fill the slot. In the submitted results, μ was set as 0.5.

4 Evaluation result

4.1 Entity linking

In this part, the results of the entity linking task are shown. We submitted two categories results for entity linking task. VSM was used to calculate the similarity after Hobbs. Based on the Hobbs algorithm, the first category way selects the result with maximum similarity score *H* and same entity type. The other way employs the combination model we designed as the following:

$$F = 0.4 * \mathrm{Sim}(S_1, S_2) + 0.4 * T + 0.2 * S, \qquad (5)$$

F is the score of a KB document. *Sim(*S_1*,* S_2*)* is from function 1. *T* is a boolean value, when the same type between target entity and KB entity, the value is set to 1. *S* is the similarity between query and document, and the similarity score is got from Indri. *Sim(*S_1*,* S_2*)* and *S* are normalized form. Then we defined a unified threshold value. When *H* or *F* is less than threshold, the result is NIL. If *F* is more than threshold, the KB with the maximum *H* or *F* value is select as the result. But as the result of the low and unified threshold value we selected, the final result in this task is much worse. The combination model evaluation result gets better result in the results we submitted. Micro-average for 3904 queries is 0.3015 and Macro-average for 560 entities is 0.2656.

After the results of entity linking task were published, we redo the experiments of the first strategy by selecting different threshold values. We divide the queries into four categories according to the number of candidate entities. We denote by *n* the number of candidate entities, *e* the entity referred by the name-mention, *Th* the threshold set for each condition, *T* the recognized type. The detailed decision rules with the VSM model are defined in table Ⅰ. In the VSM model, the greater the value is, the more similar the two documents are. The opposite case happens in the KL model. The different decision rules with the KL model are defined in table Ⅱ. We make experiments with different parts of the test set and the results are listed in table Ⅲ and table Ⅳ.

From these results, it can be seen that the KL divergence model performs a little better than the VSM model. Also, the high macro-average values demonstrate our system is effective in identifying different entities with the same mention name.

There are various sources of disadvantages with our system. The entity candidates can’t be retrieval exactly. Also, there are too many redundancies in the summary if some sentences in the original text are too long. We expect these problems can be settled by sophisticated NLP approaches.

TABLE I. DECISION RULES WITH THE VSM MODEL

n	*e (Cosine + Type)*	*Th*
n = 0	*NIL*	-
n = 1	*if Th(e) >* Th_0*, else NIL*	Th_0=0.19

n>1
&&
n<=10
	$\arg\max_e$ *Th(e) if* $\exists$*Th(e)* > Th_1 *and T(e) = T(q),*
	else $\arg\max_e$ *Th(e) if* $\exists$*Th(e)* > Th_2 *and T(e or q) = UKN,*
	else NIL
		Th_1=0.24
		Th_2=0.31

n > 10	$\arg\max_e$ *Th(e) if* $\exists$*Th(e)* > Th_3 *and T(e) = T(q),*
	else $\arg\max_e$ *Th(e) if* $\exists$*Th(e)* > Th_4 *and T(e or q) = UKN,*
	else NIL
		Th_3=0.35
		Th_4=0.35

TABLE II. DECISION RULES WITH THE KL MODEL

n	*e (KL + Type)*	*Th*
$n = 0$	*NIL*	-
$n = 1$	$if\ Th(e) < Th_0, else\ NIL$	Th_0=12.1
$n>1$ && $n<=10$	$\arg\min_e Th(e)\ if\ \exists Th(e) < Th_1\ and\ T(e) = T(q),$ $else\ \arg\min_e Th(e)\ if\ \exists Th(e) < Th_2\ and\ T(e\ or\ q) = UKN,$ $else\ NIL$	Th_1=11.1 Th_2=10.2
$n > 10$	$\arg\min_e Th(e)\ if\ \exists Th(e) < Th_3\ and\ T(e) = T(q),$ $else\ \arg\min_e Th(e)\ if\ \exists Th(e) < Th_4\ and\ T(e\ o\ r\ q) = UKN,$ $else\ NIL$	Th_3=11.1 Th_4=10.2

TABLE III. RESULTS WITH THE VSM MODEL

	Top1000	Top2000	Top3000	All
Micro-aver	0.6610	0.6540	0.6693	0.6529
Macro-aver	0.6530	0.6474	0.6966	0.7005

TABLE IV. RESUTLS WITH THE KL DIVERGENCE MODEL

	Top1000	Top2000	Top3000	All
Micro-aver	0.6790	0.6575	0.6974	0.6734
Macro-aver	0.7137	0.6684	0.7380	0.7316

4.2 Slot filling

According to the function 4, we set a threshold and then select the top slot result and top 5 slot results (for the single value slot, the top 1 is selected) above the threshold. The differences between the two sets results are not explicit. From the results, a lot of the slots are not filled correctly. The features we used do not perform the relationship very well, and the training data extracted from KB data do not math with the test data.

TABLE V. RESUTLS OF THE SLOTING FILLING

	single-slot-score	list-slot-score	SF-value score
The top 1 slot result	0.514	0.406	0.460
The top 5 slot result	0.514	0.409	0.462

References

[1] http://apl.jhu.edu/~paulmac/kbp.html

[2] http://www.lemurproject.org/indri/.

[3] http://nlp.stanford.edu/software/CRF-NER.shtml

[4] S.P. Converse. Resolving Pronominal References in Chinese with the Hobbs Algorithm. In Proceedings of the 4th SIGHAN workshop on Chinese language processing, pp. 116-122, 2005.

[5] http://nlp.stanford.edu/software/lex-parser.shtml