Jumping Distance based Chinese Person Name Disambiguation[1]

Yu Hong Fei Pei Yue-hui Yang Jian-min Yao Qiao-ming Zhu

School of Computer Science and Technology, Soochow University
No.1 Shizi street, Suzhou City, Jiansu Province, China
{hongy, 20094527004, 0727401137, jyao, qmzhu}@suda.edu.cn

Abstract

In this paper, we describe a Chinese person name disambiguation system for news articles and report the results obtained on the data set of the CLP 2010 Bakeoff-3[1]. The main task of the Bakeoff is to identify different persons from the news stories that contain the same person-name string. Compared to the traditional methods, two additional features are used in our system: 1) n-grams co-occurred with target name string; 2) Jumping distance among the n-grams. On the basis, we propose a two-stage clustering algorithm to improve the low recall.

1 Our Novel Try

For this task, we propose a Jumping-Distance based n-gram model (abbr. DJ n-gram) to describe the semantics of the closest contexts of the target person-name strings.

The generation of the DJ n-gram model mainly involves two steps. First, we mine the Jumping tree for the target string; second, we give the statistical description of the tree.

- Jumping Tree

Given a target string, we firstly extract the sentence where it locates as its closest context. Then we segment the sentence into n-grams(Chen et al. ,2009) (only Bi-gram and Tri-gram are used in this paper). For each n-gram, we regard it as the beginning of a jumping journey. And the places where we jump are the sentences which involve the n-gram. By the same way, we segment the sentences into n-grams which will be regarded as the new beginnings to open further jumping. The procedure will run iteratively until there are no sentences in the document (viz. the document which involves the target string) can be used to jump. Actually, we find there are only 3 jumps in average in our previous test and simultaneously 11 sentences in a document can be involved into the jumping journey. Thus, we can obtain a Jumping Tree where each jumping route from the initially n-gram (viz. the gram in the closes context) refer to a branch. And for each intermediate node, its child-nodes are the n-grams co-occurred with it in the same sentences.

The motivation to generate the Jumping Tree is to imitate the thinking model of human recognizing the word senses and semantics. In detail, for each intermediate node of the tree, its child-nodes all come from its closest contexts, especially the nodes co-occur with it in the same sentences which involve the real grammar and semantic relations. Thus the child-nodes normally provide the natural inference for its word sense. For example, given the string "SARS", we can deduce its sense from its child nodes "Severe", "Acute", "Respiratory" and "Syndromes" even if we see the string for the first time. On the basis, the procedure of inference run iteratively, that is, the tree always use the child nodes deduce the meaning of their father nodes then further ancestor nodes until the root. Thus the tree acts as a hierarchical understanding procedure. Additionally, the distances among nodes in the tree give the degree of semantic relation.

In the task of person-name disambiguation, we use the Jumping Tree to deduce the identities and backgrounds of a person. Each branch of the tree refers to a property of the person.

- Jumping-Distance based n-gram model

In this paper, we give a simple statistical model to describe the Jumping Tree. Given a node in the tree (viz. an n-gram), we record the

1 Supported by the National Natural Science Foundation of China under Grant No. 60970057, No.60873105.

steps jumping from the root to it, viz. the depth of the node in the tree. Then based on the priori-trained TFIDF value, we calculate the generation probability of the node as follows:

$$P = TF \cdot \frac{\alpha}{depth}$$

where the α denotes the smoothing factor.

In fact, we create more comprehensive models to describe the semantic correlations among the nodes in the Jumping Tree. The models well use the distances among the nodes in local Jumping Tree (viz. the tree generated based on the test document) and that normalized on the large-scale training data to calculate the probability of n-grams correboratively generate a semantics. They try to imitate the thinking model of human combine differents features to understand panoramic knowledge. In the task of name disambiguation, we can use the models to improve the distinguishment of different persons who have the same name. And we have illustrate the well effectiveness on the topic description and relevance measurement in other tasks, such as Link Detection. But we actually didn't use the models to perform the task of name diaambiguation this time with the aim to purely evaluate the usefulness of the Jumping Tree.

2 Systems

For the task of Chinese person name disambiguation, we submitted two systems as follows:

- System1

The system involves two main components: DJ-based name Identification error detection and DJ-based person name disambiguation.

The first component, viz. DJ-based name segmentation error detection, aims to distinguish the target string referring to person name from that referring to something else. Such as, the string "*黄海*" can be a person name "Hai Huang" but also a name of sea "the Yellow Sea". And the detection component focuses on obtaining the pure person name "Hai Huang".

The detection component firstly establish two classes of features which respectively describe the nature of human and that of things. Such as, the features "professor", "research", "honest" et al., can roughly be determined as the nature of human, and conversely the features "solid", "collapse", "deep" et al., can be that of things. For obtaining the features, we extract 10,000 documents that discuss person, eg. "Albert Einstein" and 6000 documents that discuss technology, science, geography, et.al., from Wikipedia[2]. For each document, we generate its Jumping Tree, and regard the nodes in the tree as the features. After that, we combine the weights of the same features and normalized the value by dividing that by the average weight in the specific class of features.

Based on the two classes of features, given a target string and the document where it occurs, the detection component firstly generate the Jumping Tree of the document, and then determines whether the string is person name or things by measuring the similarity of the tree to the classes of features. Here, we simply use the VSM and Cosine metric （Bagga and Baldwin, 1998） to obtain the similarity.

The second component, viz. DJ-based person name disambiguation, firstly generates the Jumping trees for all documents that involve specific person name. And a two-stage clustering algorithm is adopted to divide the documents and refer each cluster to a person. The first stage of the algorithm runs a strict division which focuses on obtaining high precision. The second stage performs a soft division which is used to improve recall. The two-stage clustering algorithm(Ikeda et al.,2009) initially obtains the optimal parameters that respectively refer to the maximum precision and recall based on training data, and then regards a statistical tradeoff as the final value of the parameters. Here, the Affinity Propagation clustering tools (Frey BJ and Dueck D, 2007) is in use.

- System2

The system is similar to the system1 except that it additionally involve Named Entity Identification (Artiles et.al,2009B; Popescu,O. and Magnini, B.,2007)before the two-stage clustering in the component of person name disambiguation. In detail, given a person name and the documents that it occurs in, the disambiguation component of System2 firstly adopt NER CRF++ toolkit[3] provided by MSRA to identify Named Entities(Chen et al., 2006) that involve the given name string, such as the entity "*李高明*" (viz. Gao-ming Li in English) when given the target name string "*高明*"(viz. Ming Gao in English). Thus the documents can be roughly

divided into different clusters of Named Entities without name segmentation errors. After that, we additionally adopt the two-stage clustering algorithm to further divide each cluster. Thus we can deal with the issue of disambiguation without the interruption of name segmentation errors.

3 Data sets

- Training dataset: They contain about 30 Chinese personal names, and a document set of about 100-300 news articles from collection of Xinhua news documents in a time span of fourteen years are provided for each personal name.
- External dataset: Chinese Wikipedia[2] personal attribution (Cucerzan, 2007; Nguyen and Cao,2008).
- Test dataset: There are about 26 Chinese personal names, which are similar to train data sets.

4 Experiments

The systems that run on test dataset are evaluated by both B-Cubed (Bagga and Baldwin, 1998; Artiles et al.,2009A) and P-IP (Artiles et al., 2007 ;Artiles et al.,2009A). And the systems that run on training dataset were only evaluated by B-Cubed.

In experiments, we firstly evaluate the performance of name segmentation error detection on the training dataset. For comparison, we additionally perform another detection method which only using Name Entity Identifcation (NER CRF++ tools) to distinguish name-strings from the discarded ones. The results are shown in table 1. We can find that our error detection method can achieve more recall than NER, but lower precision.

Besides, we evaluate the performance of the two-stage clustering in the component of name disambiguation step by step. Four steps are in use to evaluate the first-stage clustering method as follows:

- DJ^2

This step look like to run the system1 mentionedin in section 3 which don't involve the prior-division of documents by using NER before the first-stage clustering in the component of name disambiguation. Especially it don't perform the second-stage clustering to improve the recall probability.

- DJ^2+NER

This step is similar to the step of DJ^2 mentioned above except that it perform the prior-divison of documents by using NER.

- NER+DJ

This step is also similar to the step of DJ^2 except that its name segmentation error detection performs by using the NER.

- NER^2+DJ

This step is similar to the step of NER+DJ except that it involve the treatment of prior-divison as that in DJ^2+NER.

The performances of the four steps are shown in table 2. We can find that all steps achieve poor recall. And the step of DJ^2 achieve the best F-score although it don't involve the prior-division. That is because NER is helpful to improve precision but not recall, as shown in table 1. Conversely, DJ^2 can avoid the bias caused by the procedure of greatly maximizing the precision.

	P	**recall**	**F-score**
DJ-based	0.62	0.81	0.70
NER-based	0.91	0.77	0.71

Table 1: Performance of name segmentation error detection

	P	**IP**	**F-score**
DJ^2	80.49	53.85	60.12
DJ^2+NER	88.56	51.30	59.02
NER+DJ	93.27	46.78	57.44
NER^2+DJ	97.79	42.13	55.47

Table 2: Performances of the-stage clustering

Additionally, another two steps are used to evluate the both two stages of clustering in name disambiguation. The steps are as follows:

- DJ^2+NER_2

This step is similar to the step of DJ^2+NER except that it additionally run the second-stage clustering to improve recall.

- NER^2+DJ_2

This step also run the second-stage clustering on the basis of NER^2+DJ.

The performances of the two step are shown in table 3. We can find that the F-scores both have been improved substantially. And the two

steps still maintain the original distribution between precision and recall. That is, the DJ^2+NER_2, which has outperformance on recall in the name segmentation error detection, still maintain the higher recall at the second-stage clustering. And NER^2+DJ_2 also maintains higher precision. This illustrates that the clustering has no ability to remedy the shortcomings of NER in the prior-division.

	P	IP	F-score
DJ^2+NER_2	82.65	63.40	66.59
NER^2+DJ_2	87.71	60.45	66.23

Table 3: Performances of two-stage clustering

The test results of the two systems mentioned in section 3 are shown in the table 4. We also show the performances of each stage clustering as that on training dataset. We can find that the poor performance mainly come from the low recall, which illustrates that the DJ-based n-gram disambiguation is not robust.

	B-Cubed		
	precision	recall	F-Score
System1(one	85.26	28.43	37.74
System1(both	84.51	44.17	51.42
	P-IP		
	P	IP	F-Score
System2(one	88.4	39.47	50.52
System2(both	88.36	55.23	63.89

Table 4 :Test results

5.Conclusions

In this paper, we report a hybrid Chinese personal disambiguation system and a novel algorithm for extract useful global n-gram features from the context .Experiment showed that our algorithm performed high precision and poor recall. Furthermore, two-stage clustering can handl a change in the one-stage clustering algorithm, especially for recall score. In the future, we will investigate global new types of features to improve the recall score and local new types of features to improve the precision score. For instance, the location and organization besides the person in the named-entities. And we try to use Hierarchical Agglomerative Clustering algorithm to help raise the recall score.

References

Artiles J, J Gonzalo and S Sekine. 2007. The SemEval-2007 WePS Evaluation: "Establishing a benchmark for the Web People Search Task.", The SemEval-2007, 64-69, Association for Computational Linguistics.

Artiles Javier, Julio Gonzalo and Satoshi Sekine.2009A. "WePS 2 Evaluation Campaign: overview of the Web People Search Clustering Task," In 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference.

Artiles J, E Amig´o and J Gonzalo. 2009B.The Role of Named Entities in Web People Search. Proceedings of the 2009 Conference on Empirical Methods Natural Language Processing, 534–542,Singapore, August 2009.

Bagga A and Baldwin B. 1998. Entity-based cross-document coreferenceing using the Vector Space Model.Proceedings of the 17th international conference on computational linguistics. Volume 1, 79-85.

Chen,Ying., Sophia Yat., Mei Lee and Chu-Ren Huang. 2009. PolyUHK:A Roubust Information Extraction System for Web Personal Names In 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference.

Chen Wen-liang, Zhang Yu-jie. 2006. Chinese Named Entity Recognition with Conditional Random Fields. Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing.

Cucerzan, Silviu. 2007. Large scale named entity Disambiguation based on Wikipedia data. In The EMNLP-CoNLL-2007.

Frey BJ and Dueck D. 2007. Clustering by Passing Messages Between Data Points .science, 2007 - sciencemag.org.

Ikeda MS, Ono I, Sato MY and Nakagawa H. 2009. Person Name disambiguation on the Web by Two-Stage Clustering. In 2nd Web People Search Evaluation Workshop(WePS 2009),18th WWW Conference.

Popescu,O and Magnini, B. 2007. IRST-BP:Web People Search Using Name Entities.Proceeding s of the 4th International Workshop on Semantic Evaluations (SemEval-2007), 195-198, Prague June 2007. Association for Computational Linguistics.