DLUT: Chinese Personal Name Disambiguation with Rich Features

Dongliang Wang
Department of Computer Science and Engineering, Dalian University of Technology
wdl129@163.com

Degen Huang
Department of Computer Science and Engineering, Dalian University of Technology
huangdg@dlut.edu.cn

Abstract

In this paper we describe a person clustering system for a given document set and report the results we have obtained on the test set of Chinese personal name (CPN) disambiguation task of CIPS-SIGHAN 2010. This task consists of clustering a set of Xinhua news documents that mention an ambiguous CPN according to named entity in reality. Several features including named entities (NE) and common nouns generated from the documents and a variety of rules are employed in our system. This system achieves F = 86.36% with B_Cubed scoring metrics and F = 90.78% with purity_based metrics.

1 Introduction

As the amount of web information expands at an ever more rapid pace, extraction of information for specific named entity is more and more important. Usually there are named-entity ambiguity in web data, for example more than one person use a same name, therefore it is difficult to decide which document refers to a specific named entity.

The goal of CPN disambiguation is to clustering input Xinhua news corpus by the entity each document refers to. The new documents which span a time of fourteen years are extracted on web.

As description of CPN disambiguation task of CIPS-SIGHAN 2010, Chinese personal name disambiguation is potentially more challenging due to the need for word segmentation, which could introduce errors that can in large part be avoided in the English task.

In this paper we employ a CPN disambiguation system that extracts NE and common nouns from the input corpus as features, and then computes the similarity of each two documents in the corpus based on feature vector. Hierarchical Agglomerative Clustering (HAC) algorithm (AK Jain et al., 1999) is used to implement clustering.

After a great deal of analysis of news corpus, we constitute several rules, the experiments show that these rules can improve the result of this task.

The remainder of this paper is organized as follows. Section 2 introduces the preprocessing of test corpus, and in section 3 we present the methodology of our system. In section 4 we present the experimental results and give a conclusion in section 5.

2 Preprocessing

In this step, we mainly complete the works as follows.

Firstly, corpuses including a given name string are in different files, one document one file. In order to convenient for processing, we combine these documents into one file, distinguish them with document id.

Secondly, some news corpuses have several subtitles but usually only part of them including focused name string, the others are noise of disambiguate of focused named entity, for example a news about sports may contain several subtitles about basketball, swimming, race and so on. These noises are removed from the corpus by us.

Lastly, there is a lack of date-line in a few documents; in general, these data-lines are recognized as part of text, they can be recognized through simple matching method. Because data-lines have consistent format as “新华社**月*日电”.

3 Methodology

The system follows a procedure include: word segmentation, the detection of ambiguous objects, feature extractions, computation of document similarity and clustering.

First, the text is segmented by a word segmentation system explored by Luo and Huang (2009). The second step is extract all features from segmented text, all features are put into two feature vectors: NE vector and common noun vector. Then we will compute the distance between corresponding vectors of each two documents, the standard SoftTFIDF (Chen and Martin, 2007) are employed to compute the distance between two feature vectors. Lastly, we use the HAC algorithm for clustering of documents.

3.1 Word Segmentation

Word segmentation is a base and difficult work of natural language processing (NLP) and a precondition of feature extraction. In this paper, the word segmentation system explored by Luo and Huang (2009) are employed to do this work.

This system training on the corpus of 2000's "People's Daily". In addition, this system can recognize named entities including personal name, location name and organization name. We can extract these NEs by part-of-speech (POS) directly.

3.2 The Detection of Ambiguous Entities

Given a name string, the documents can be divided into three groups:

(1) Documents which contain names that are exactly match the query name string.

(2) Documents which contain names that have a substring exactly match the query name string.

(3) Documents which contain the query name string that is not personal name.

After word segmentation, all personal names are labeled by system, when we find one personal name or its substring match the query name string; we will cluster this document according to the name. If we failure all over the document, it's considered that this document belong to category (3), it will be discarded.

The ambiguous personal name in a document may refer to multiple entities, for example a news about party of namesakes, but this is a very small probability event, so we assume that all mentions in one document refers to the same entity, viz. "one person one document".

Although we assume that "one person one document", the same personal name may occur more than once. Some times the word segmentation system will give the same personal name different labels in one document, for example a personal name "杨永强" may be recognized as "杨永" and "杨永强" in different sentence in one document. Suppose that $P_1, P_2, \dots, P_n$ are recognized names that match the query name string, $T_1, T_2, \dots, T_n$ are the corresponding occur times. We use the following method to ensure the final needed personal name:

(1) If $T_i > T_j$ for $j = 1, 2, \dots, i-1, i+1, \dots, n$, P_i is selected as the final needed personal name, else go to step (2).

(2) Define $S = \{ T_1, T_2, \dots, T_n \}$, $E_1 = \{T_{11}, T_{12}, \dots, T_{1m}\}$, $E_2 = S - E1$ satisfying $T_{11} = T_{12} = \dots = T_{1m}$, $E_1 \subseteq S$ and $T_i > T_j$ ($T_i \in E_1, T_j \in E_2$). F_i shows the word before P_i and B_i after P_i. For each $T_i \in E_1$, connect F_i, T_i and B_i into a new string named R_i, we can get $R = \{R_{11}, R_{12}, \dots, R_{1m}\}$ corresponding to E1, the longest common substring of R are considered the final needed personal name.

3.3 Features

We define local sentence as sentences which contain the query name string, the features extracted from local sentences named local features. Otherwise, all sentences except local sentences in a document are named global sentences; the features extracted from global sentences are global features. The reason to distinguish them is because they have different contribution to similarity computation. Local features are generally considered more important than global features, therefore a high weight should be given to local features.

Named entities are important information about focused name. In this paper, NEs include personal names, location names and organization names. Location name and organization name usually indicate the region and department of focused name, and personal names usually have high co-occurrence rate, for example "邓亚萍" and "高军" are two names of table tennis players, so they always appear in a same news document

about table tennis. The NE features which have been tagged by segmentation system can be extracted from the document directly.

We also consider the features of common nouns. Semantically independent common nouns such as person's job and person's hobby etc usually include some useful information about the ambiguous object. We attempt to capture these noun features and use them as elements in feature vector.

Location names in data-line. The location name in the data-line indicates the place the news had occurred, if two documents have the same date-line location name, and then there is a good chance that these two documents refer the same person.

Appellation of query name. Appellation usually demonstrate a person's identity, for example, if the appellation of the query name is "记者", it shows that he or she is a journalist. As location names in data-line, if two query names have the same appellation, the possibility of them refer to the same person increased. The word segmentation system doesn't clearly marked out appellation but marked as common noun. In generally, appellations appear neighbor in front of name, so we collect the common nouns neighbor front of query names as their appellations.

So far, we have developed four feature vectors: local NE vector, local common noun vector, global NE vector and global common noun vector. Given feature vectors, we need to find a way to learn the similarity matrix. In this paper, we choose the standard TF-IDF method to calculate the similarity matrix. Location name in date-line and appellation of query name will be used in rule method without similarity calculation.

3.4 Similarity Matrix

Given a pair of feature vectors consisting of NEs or common nouns, we need to choose a similarity scheme to calculate the similarity matrix. The standard TF-IDF method is introduced here, then a little change for Chinese string.

Standard TF-IDF: Given a pair of vector S and T, S = (s1, s2, …, sn), T = (t1, t2, …, tm). Here, si (i = 1, …, n) and tj (j = 1, …, m) are NE or common noun. We define:

$$CLOSE(\theta;S;T)= \{w;w\in S,\exists v\in T,dist(w,v)>\theta\} \quad (1)$$

Where dist(w;v) is the Jaro-Winkler distance function (Winkler, 1999), which will be introduced later.

$$D(w;T)=\max_{v\in T} dist(w;v) \quad (2)$$

Then the standard TF-IDF SoftTFIDF is computed as:

$$SoftTFIDF(S,T)= \sum_{w\in CLOSE(\theta;S;T)} V(w,S)*V(w,T)*D(w,T) \quad (3)$$

$$V(w,S)=\frac{V'(w,S)}{\sqrt{\sum_{w\in S} V'(w,S)^2}} \quad (4)$$

$$V'(w,S)=\log(TF_{w,S}+1)*\log(IDF_w) \quad (5)$$

Where $TF_{w,S}$ is the frequency of substring w in S, and IDF_w is the inverse of the fraction of documents in the corpus that contain w. Suppose N_t is total number of documents, N_w is total number of documents which contain w. Then IDF_w computed as:

$$IDF_\omega=\frac{N_t}{N_w} \quad (6)$$

The Jaro-Winkler distance Jw of two given strings s1 and s2 as shown in formula (7), l is the length of common prefix at the start of the string up to a maximum of 4 characters, p is a constant scaling factor for how much the score is adjusted upwards for having common prefixes, the value for p is 0.1.

$$d_w=d_j+lp(1-d_j) \quad (7)$$

$$d_j=\left(\frac{m}{|s1|}+\frac{m}{|s2|}+\frac{m-t}{m}\right) \quad (8)$$

In formula (8) m is the number of matching characters, t is the number of transpositions. In order to be consistent with the English strings, a Chinese character is seen as two English characters.

Corresponding to four feature vectors, we can calculate the four similarities: S(gNE), S(gCN), S(lNE), S(lCN). The similarity between two documents (DS) is computed as:

$$DS=\lambda*\frac{S(lNE)+S(gNE)}{2}+(1-\lambda)*\frac{S(lCN)+S(gCN)}{2} \quad (9)$$

As time is tight, we just give λ a value of 0.8 with out experiment because we consider NEs have stronger instructions.

3.5 Clustering

Clustering is a key work of this task, it is very important to choose a clustering algorithm. Here we use HAC algorithm to do clustering. HAC algorithm is an unsupervised clustering algorithm, which can be described as follows:

(1) Initialization. Every document is regarded as a separate class.

(2) Repetition. Computing the similarity of each of the two classes, merge the two classes whose similarity are the highest and higher than the threshold value of δ into a new class.

(3) Termination. Repeat step (2) until all classes don't satisfy the clustering condition.

Suppose document class F = $\{f_1, f_2, \ldots, f_n\}$ and K = $\{k_1, k_2, \ldots, k_m\}$, f_i and k_j are documents in class F and class K, then the similarity between F and K is:

$$S(J,K) = \frac{\sum_{i,j} S(f_i, k_j)}{m * n} \quad (9)$$

If two documents have different query name, obviously they refer to different person, only documents which have same query name will be clustered. Before clustering, several rules are afforded to improve the clustering condition. These rules are generally applicable to news corpus.

(1) If two documents have the same query name and both of them are reporter, and both date-lines have the same location name, then combine the two documents into one class.

(2) If two documents have the same query name and another same personal name, then combine the two documents into one class.

(3) If two documents have the same query name and both date-lines have the same location name, then double the similarity, else halve the similarity.

(4) If two documents have the same query name and both personal names have the same appellation, then double the similarity, else halve the similarity.

4 Evaluation

In order to prove the validity of the rule approach, a group of experiments are performed on the train set of Chinese personal name disambiguation task of CIPS-SIGHAN 2010. The result is shown in Table 1. R1 is the result without rules, and R2 shows the accuracy after adding the rules.

The system performance on the test set of CPN disambiguation task of CIPS-SIGHAN 2010 is F = 90.78% evaluated with P_IP evaluation, and F = 86.36% with B_Cubed evaluation. The accuracy is shown in Table 2.

B_Cubed	Precision	Recall	F
R1	70.56	86.77	74.74
R2	78.05	84.99	79.60
P_IP	Purity	Inverse Purity	F
R1	77.22	90.48	81.20
R2	82.92	88.30	84.29

Table 1. Experimental results for system with rules and without rules on training set

B_Cubed	Precision	Recall	F
	82.96	91.33	86.36
P_IP	Purity	Inverse Purity	F
	87.94	94.21	90.78

Table 2. The results on test set

5 Conclusion

We described our system that disambiguates Chinese personal names in Xinhua corpus. We mainly focus on extracting rich features from documents and computing the similarity of each two documents. Several rules are introduced to improve the accuracy and have proved effective.

References

Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. 1999. Data clustering: A review. ACM Computing Surveys, 31(3): 264-323.

Bradley Malin. 2005. *Unsupervised Name Disambiguation via Network Similarity.* In proceedings SIAM Conference on Data Mining, 2005.

Chen Ying, James Martin. 2007. *CU-COMSEM: Exploring Rich Features for Unsupervised Web Personal Name Disambiguation.* In proceedings of Semeval 2007, Association for Computational Linguistics, 2007.

Chen Ying, Sophia Y. M. Lee and Churen Huang. 2009. *PolyUHK:A Robust Information Extraction System for Web Personal Names.* In proceedings of Semeval 2009, Association for Computational Linguistics, 2009.

Gusfield, Dan. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK

Javier Artiles, J. Gonzalo and S. Sekine. WePS2 Evaluation Campaign: Overview of the Web People Search Clustering Task. In proceedings of Semeval 2009, Association for Computational Linguistics, 2009.

Luo Yanyan, Degen Huang. 2009. *Chinese word segmentation based on the marginal probabilities Generated by CRFs*. Journal of Chinese Information Processing, 23(5): 3-8.

Octavian Popescu, B. Magnini. 2007. *IRST-BP: Web People Search Using Name Entities.* In proceedings of Semeval 2007, Association for Computational Linguistics, 2007.

William E. Winkler. 1999. The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04.