Clustering Technique in Multi-Document Personal Name Disambiguation

Chen Chen
Key Laboratory of Computational Linguistics (Peking University),
Ministry of Education, China
chenchen@pku.edu.cn

Hu Junfeng
Key Laboratory of Computational Linguistics (Peking University),
Ministry of Education, China
hujf@pku.edu.cn

Wang Houfeng
Key Laboratory of Computational Linguistics (Peking University),
Ministry of Education, China
wanghf@pku.edu.cn

Abstract

Focusing on multi-document personal name disambiguation, this paper develops an agglomerative clustering approach to resolving this problem. We start from an analysis of pointwise mutual information between feature and the ambiguous name, which brings about a novel weight computing method for feature in clustering. Then a trade-off measure between within-cluster compactness and among-cluster separation is proposed for stopping clustering. After that, we apply a labeling method to find representative feature for each cluster. Finally, experiments are conducted on word-based clustering in Chinese dataset and the result shows a good effect.

1 Introduction

Multi-document named entity co-reference resolution is the process of determining whether an identical name occurring in different texts refers to the same entity in the real world. With the rapid development of multi-document applications like multi-document summarization and information fusion, there is an increasing need for multi-document named entity co-reference resolution. This paper focuses on multi-document personal name disambiguation, which seeks to determine if the same name from different documents refers to the same person.

This paper develops an agglomerative clustering approach to resolving multi-document personal name disambiguation. In order to represent texts better, a novel weight computing method for clustering features is presented. It is based on the pointwise mutual information between the ambiguous name and features. This paper also develops a trade-off point based cluster-stopping measure and a labeling algorithm for each clusters. Finally, experiments are conducted on word-based clustering in Chinese dataset. The dataset contains eleven different personal names with varying-sized datasets, and has 1669 texts in all.

The rest of this paper is organized as follows: in Section 2 we review the related work; Section 3 describes the framework; section 4 introduces our methodologies including feature weight computing with pointwise mutual information, cluster-stopping measure based on trade-off point, and cluster labeling algorithm. These are the main contribution of this paper; Section 5 discusses our experimental result. Finally, the conclusion and suggestions for further extension of the work are given in Section 6.

2 Related Work

Due to the varying ambiguity of personal names in a corpus, existing approaches typically cast it as an unsupervised clustering problem based on vector space model. The main difference among these approaches lies in the features, which are used to create a similarity space. Bagga & Baldwin (1998) first performed within-document co-reference resolution, and then explored features in local context. Mann & Yarowsky (2003) extracted local biographical information as features. Al-Kamha and Embley (2004) clustered search results with feature set including attributes, links and page similarities. Chen and Martin (2007) explored the use of a range of syntactic and semantic features in unsupervised clustering of documents. Song (2007) learned the PLSA and LDA model as feature sets. Ono *et al.* (2008) used mixture features including co-occurrences

Proceedings of the ACL-IJCNLP 2009 Student Research Workshop, pages 88–95,
Suntec, Singapore, 4 August 2009.

of named entities, key compound words, and topic information. Previous works usually focus on feature identification and feature selection. The method to assign appropriate weight to each feature has not been discussed widely.

A major challenge in clustering analysis is determining the number of 'clusters'. Therefore, clustering based approaches to this problem still require estimating the number of clusters. In Hierarchy clustering, it equates to determine the stopping step of clustering. The measure to find the "knee" in the criterion function curve is a well known cluster-stopping measure. Pedersen and Kulkarni had studied this problem (Pedersen and Kulkarni, 2006). They developed cluster-stopping measures named PK1, PK2, PK3, and presented the Adapted Gap Statistics.

After estimating the number of 'clusters', we obtain the clustering result. In order to label the 'clusters', the method that finding representative features for each 'cluster' is needed. For example, the captain John Smith can be labeled as captain. Pedersen and Kulkarni (2006) selected the top N non-stopping word features from texts grouped in a cluster as label.

3 Framework

On the assumption of "one person per document" (i.e. all mentions of an ambiguous personal name in one document refer to the same personal entity), the task of disambiguating personal name in text set intends to partition the set into subsets, where each subset refer to one particular entity.

Suppose the set of texts containing the ambiguous name is denoted by $D= \{d_1,d_2,\ldots,d_n\}$, and d_i $(0<i<n+1)$ stands for one text. The entities with the ambiguous name are denoted by a set $E= \{e_1,e_2,\ldots,e_m\}$, where the number of entities 'm' is unknown. The ambiguous name in each text d_i indicates only one entity e_k. The aim of the work is to map an ambiguous name appearing in each text to an entity. Therefore, those texts indicating the same entity need to be clustered together.

In determining whether a personal name refers to a specific entity, the personal information, social network information and related topics play important roles, all of which are expressed by words in texts,. Extracting words as features, this paper applies an agglomerative clustering approach to resolving name co-reference. The framework of our approach consists of the following seven main steps:

Step 1: Pre-process each text with Chinese word segmentation tool;

Step 2: Extract words as features from the set of texts D;.

Step 3: Represent texts $d_1,\ldots,d_n$ by features vectors;

Step 4: Calculate similarity between texts;

Step 5: Cluster the set D step by step until only one cluster exists;

Step 6: Estimate the number of entities in accordance with cluster-stopping measure;

Step 7: Assign each cluster a discriminating label.

This paper focuses on the *Step 4*, *Step 6* and *Step 7*, i.e., feature weight computing method, clustering stopping measure and cluster labeling method. They will be described in the next section in detail.

Step1 and *Step3* are simple, and there is no further description here. In *Step 2*, we use co-occurrence words of the ambiguous name in texts as features. In the process of agglomerative clustering (see *Step 5*), each text is viewed as one cluster at first, and the most similar two clusters are merged together as a new cluster at each round. After replacing the former two clusters with the new one, we use average linked method to update similarity between clusters.

4 Methodology

4.1 Feature weight

Each text is represented as a feature vector, and each item of the vector represents the weight value for corresponding feature in the text. Since our approach is completely unsupervised we cannot use supervised methods to select significant features. Since the weight of feature will be adjusted well instead of feature selection, all words in set *D* are used as feature in our approach.

The problem of computing feature weight is involved in both text clustering and text classification. By comparing the supervised text classification and unsupervised text clustering, we find that the former one has a better performance owing to the selection of features and the computing method of feature weight. Firstly, in the application of supervised text classification, features can be selected by many methods, such as, Mutual Information (MI) and Expected Cross Entropy (ECE) feature selection methods. Secondly, model training methods, such as SVM model, are generally adopted by programs when to find the

optimal feature weight. There is no training data for unsupervised tasks, so above-mentioned methods are unsuitable for text clustering.

In addition, we find that the text clustering for personal name disambiguation is different from common text clustering. System can easily judge whether a text contains the ambiguous personal name or not. Thus the whole collection of texts can be easily divided into two classes: texts with or without the name. As a result, we can easily calculate the pointwise mutual information between feature words and the personal name. To a certain extent, it represents the correlative degree between feature words and the underlying entity corresponding to the personal name.

For these reasons, our feature weight computing method calculates the pointwise mutual information between personal name and feature word. And the value of pointwise mutual information will be used to expresse feature word's weight by combining the feature's *tf (*the abbreviation for term-frequency*)* in text and *idf (*the abbreviation for inverse document frequency*)* in dataset. The formula of feature weight computing proposed in this paper is as below, and it is need both texts containing and not containing the ambiguous personal name to form dataset *D*. For each t_k in d_i that contains *name*, its *mi_weight* is computed as follow:

$$\begin{aligned}&\text{mi\_weight}(t_k, name, d_i) = (1+\log(tf(t_k, d_i)))\\&\times\log(1+\text{MI}(t_k, name))\times\log(|D|/df(t_k))\end{aligned} \tag{1}$$

And

$$\begin{aligned}\text{MI}(t_k, name) &= \frac{p(name, t_k)}{p(name)\times p(t_k)}\\&= \frac{df(name, t_k)/|D|}{df(name)\times df(t_k)/|D|^2}\\&= \frac{df(name, t_k)\times|D|}{df(name)\times df(t_k)}\end{aligned} \tag{2}$$

Where t_k is a feature; *name* is the ambiguous name; d_i is the i^{th} text in dataset; $tf(t_k,d_i)$ represents term frequency of feature t_k in text d_i; $df(t_k)$, *df(name)* is the number of the texts containing t_k or *name* in dataset *D* respectively; $df(t_k,name)$ is the number of texts containing both t_k and *name*; $|D|$ is the number of all the texts.

Formula (2) can be comprehended as: if word t_k occurs much more times in texts containing the ambiguous name than in texts not containing the name, it must have some information about the name.

A widely used approach for computing feature weight is *tf*idf* scheme as formula (3) (Salton and Buckley. 1998), which only uses the texts containing the ambiguous name. We denote it by *old_weight* . For each t_k in d_i containing *name,* the *old_weight* is computed as follow:

$$\begin{aligned}&\text{old\_weight}(t_k, name, d_i)\\&= (1+\log(tf(t_k, d_i)))\\&\times\log(df(name)/df(t_k, name))\end{aligned} \tag{3}$$

The first term on the right side is *tf*, and the second term is *idf*. If the *idf* scheme is computed in the whole dataset *D* for reducing noise, the weight computing formula can be expressed as follow, and is denoted by *imp_weight*:

$$\begin{aligned}&\text{imp\_weight}(t_k, d_i)\\&= (1+\log(tf(t_k, d_i)))\times\log(|D|/df(t_k))\end{aligned} \tag{4}$$

Before clustering, the similarity between texts is computed by cosine value of the angle between vectors (such as $\mathbf{d_x}$, $\mathbf{d_y}$ in formula (5)):

$$\cos(\mathbf{d}_x, \mathbf{d}_y) = \frac{\mathbf{d}_x\cdot\mathbf{d}_y}{\|\mathbf{d}_x\|\cdot\|\mathbf{d}_y\|} \tag{5}$$

Each item of the vector (i.e. $\mathbf{d_x}$, $\mathbf{d_y}$) represents the weight value for corresponding feature in the text.

4.2 Cluster-stopping measure

The process of clustering will produce *n* cluster results, one for each step. Independent of clustering algorithm, the cluster stopping measure should choose the cluster results which can represent the structure of data.

A fundamental and difficult problem in cluster analysis is to measure the structure of clustering result. The geometric structure is a representative method. It defines that a "good" clustering results should make data points from one cluster "compact", while data points from different cluster are "separate" as far as possible. The indicators should quantify the "compactness" and "separation" for clusters, and combine both. In the study of cluster stopping measures by Pedersen and Kulkarni (2006), the criterion functions defines text similarity based on cosine value of the angle between vectors. Their cluster-stopping measures focused on finding the 'knee' of criterion function.

Our cluster-stopping measure is also based on the geometric structure of dataset. The measure aims to find the trade-off point between within-cluster compactness and among-cluster separation. Both the within-cluster compactness (Internal critical function) and among-cluster

separation (External critical function) are defined by Euclidean distance. The hybrid critical function (Hybrid critical function) combines internal and external criterion functions.

Suppose that the given dataset contains *N* references, which are denoted as: $d_1,d_2,\ldots,d_N$; the data have been repeatedly clustered into *k* clusters, where $k=N,\ldots,1$; and clusters are denoted as C_r, $r=1,\ldots k$; and the number of references in each cluster is n_r, so $n_r=|C_r|$. We introduce *Incrf* (Internal critical function), *Excrf* (External critical function) and *Hycrf* (Hybrid critical function) to measure it as follows.

$$\mathrm{Incrf}(k)=\sum_{i=1}^{k}\sum_{\mathbf{d_x},\mathbf{d_y}\in\mathbf{C_i}}\left\|\mathbf{d_x}-\mathbf{d_y}\right\|^2 \quad (6)$$

$$\mathrm{Excrf}(k)=\sum_{i=1}^{k}\sum_{j=1,\,j\neq i}^{k}\frac{1}{n_i n_j}\sum_{\mathbf{d_x}\in\mathbf{C_i},\mathbf{d_y}\in\mathbf{C_j}}\left\|\mathbf{d_x}-\mathbf{d_y}\right\|^2 \quad (7)$$

$$\mathrm{Hycrf}(k)=\frac{1}{\mathrm{M}}\times(\mathrm{Incrf}(k)+\mathrm{Excrf}(k)) \quad (8)$$

Where M=Incrf(1)=Excrf(N)

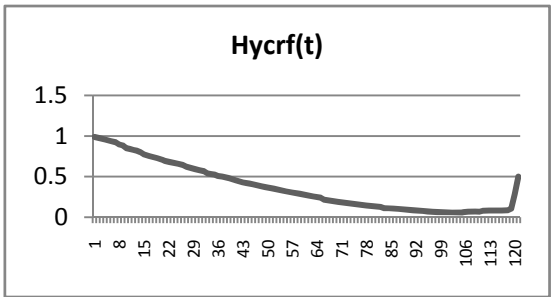

Figure 1 *Hycrf* vs. *t* (N-k)

Chen proved the existence of the minimum value between (0,1) in Hycrf(k) (see Chen *et al*. 2008). The *Hycrf* value in a typical Hycrf(t) curve is shown as Figure 1, where t=N-k.

Function *Hycrf* based on *Incrf* and *Excrf* is used as the Hybrid criterion function. The *Hycrf* curve will rise sharply after the minimum, indicating that the cluster of several optimal partitions' subsets will lead to drastic drop in cluster quality. Thus cluster partition can be determined. Using the attributes of the *Hycrf(k)* curve, we put forward a new cluster-stopping measure named trade-off point based cluster-stopping measure (TO_CSM).

$$\mathrm{TO\_CSM}(k)=\frac{1}{\mathrm{Hycrf}(k+1)}\times\frac{\mathrm{Hycrf}(k)}{\mathrm{Hycrf}(k+1)} \quad (9)$$

Trade-off point based cluster-stopping measure (*TO_CSM*) selects the *k* value which maximizes *TO_CSM(k)*, and indicates the number of cluster. The first term on the right side of formula (9) is used to minimize the value of *Hycrf(k)*, and the second one is used to find the 'knee' rising sharply.

4.3 Labeling

Once the clusters are created, we label each entity to represent the underlying entity with some important information. A label is represented as a list of feature words, which summarize the information about cluster's underlying entity.

The algorithm is outlined as follows: after clustering *N* references into *m* clusters, for each cluster C_k in $\{C_1, C_2, \ldots, C_m\}$, we calculate the score of each feature for C_k and choose features as the label of C_k whose scores rank top N. In particular, the score caculated in this paper is different from Pedersen and Kulkarni's (2006). We combine pointwise mutual information computing method with term frequency in cluster to compute the score.

The formula of feature scoring for labeling is shown as follows:

$$\begin{aligned}\mathrm{Score}(t_k,C_i)&=\mathrm{MI}(t_k,name)\times\mathrm{MI}_{\mathrm{name}}(t_k,C_i)\\&\quad\times(1+\log(tf(t_k,C_i)))\end{aligned} \quad (10)$$

The calculation of $MI(t_k,name)$ is shown as formula (2) in subsection 4.1. $tf(t_k,C_i)$ represents the total occurrence frequency of feature t_k in cluster C_i . The $MI_{name}(t_k,C_i)$ is computed as formula (11):

$$\begin{aligned}\mathrm{MI}_{name}(t_k,\mathrm{C}_i)&=\frac{p(t_k,C_i)}{p(t_k)\times p(C_i)}\\&=\frac{df(t_k,C_i)/|D|}{df(t_k)\times df(C_i)/|D|^2}\\&=\frac{df(t_k,C_i)\times|D|}{df(t_k)\times df(C_i)}\end{aligned} \quad (11)$$

In formula (10), the weight of stopping words can be reduced by the first item. The second item can increase the weight of words with high distinguishing ability for a certain ambiguous name. The third item of formula (10) gives higher scores to features whose frequency are higher.

5 Experiment

5.1 Data

The dataset is from WWW, and contains 1,669 texts with eleven real ambiguous personal names. Such raw texts containing ambiguous names are collected via search engine[1], and most of them are news. The eleven person-names are, "刘易斯 Liu-Yi-si 'Lewis'", "刘淑珍 Liu-Shu-zhen ", "李强 Li-Qiang", "李娜 Li-Na", "李桂英 Li-Gui-ying", "米歇尔 Mi-xie-er 'Michelle'", "玛丽 Ma-Li 'Mary'", "约翰逊 Yue-han-xun 'Johnson'", "王涛 Wang-Tao", "王刚 Wang-Gang", "陈志强 Chen-Zhi-qiang". Names like "Michelle", "Johnson" are transliterated from English to Chinese, while names like "Liu –Shu-zhen", "Chen-Zhi-qiang" are original Chinese personal names. Some of these names only have a few persons, while others have more persons.

Table 1 shows our data set. "#text" presents the number of texts with the personal name. "#per" presents the number of entities with the personal name in text dataset. "#max" presents the maximum of texts for an entity with the personal name, and "#min" presents the minimum.

	#text	#per	#max	#min
Lewis	120	6	25	10
Liu-Shu-zhen	149	15	28	3
Li-Qiang	122	7	25	9
Li-Na	149	5	39	21
Li-Gui-ying	150	7	30	10
Michelle	144	7	25	12
Mary	127	7	35	10
Johnson	279	19	26	1
Wang-Gang	125	18	26	1
Wang-Tao	182	10	38	5
Chen-Zhi-qiang	122	4	52	13

Table 1 Statistics of the test dataset

We first convert all the downloaded documents into plain text format to facilitate the test process, and pre-process them by using the segmentation toolkit ICTCLAS[2].

In testing and evaluating, we adopt B-Cubed definition for *Precision*, *Recall* and *F-Measure* as indicators (Bagga, Amit and Baldwin. 1998). *F-Measure* is the harmonic mean of *Precision* and *Recall*.

The definitions are presented as below:

$$precision = \frac{1}{N}\sum_{d \in D} precision_d \quad (12)$$

$$recall = \frac{1}{N}\sum_{d \in D} recall_d \quad (13)$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \quad (14)$$

where $precision_d$ is the precision for a text d. Suppose the text d is in subset A, $precision_d$ is the percentage of texts in A which indicates the same entity as d. $Recall_d$ is the recall ratio for a text d. $Recall_d$ is the ratio of number of texts which indicates the same entity as d in A to that in corpus D. $n = |D|$, D refers to a collection of texts containing a particular name (such as Wang Tao, e.g. a set of 200 texts, n = 200). Subset A is a set formed after clustering (text included in class), and d refers to a certain text that containing "Wang Tao".

5.2 Result

All the 1669 texts in the dataset are employed during experiment. Each personal name disambiguation process only clusters the texts containing the ambiguous name. After pre-processing, in order to verify the *mi_weight* method for feature weight computing, all the words in texts are used as features.

Using formula (1), (3) and (4) as feature weight computing formula, we can get the evaluation of cluster result shown as table 2. In this step, cluster-stopping measure is not used. Instead, the highest F-measure during clustering is highlighted to represent the efficiency of the feature weight computing method.

Further more, we carry out the experiment on the trade-off point based cluster-stopping measure, and compare its cluster result with highest F-measure and cluster result determined by cluster-stopping measure PK3 proposed by Pedersen and Kulkarni's. Based on the experiment in Table 2, a structure tree is constructed in the clustering process. Cluster-stopping measures are used to determine where to stop cutting the dendrogram. As shown in Table 3, the TO-CMS method predicts the optimal results of four names in eleven, while PK3 method predicts the optimal result of one name, which are marked in a bold type.

[1] April.2008
[2] http://ictclas.org/

	old_weight			imp_weight			mi_weight		
	#pre	**#rec**	**#F**	**#pre**	**#rec**	**#F**	**#pre**	**#rec**	**#F**
Lewis	0.9488	0.8668.	0.9059	1	1	1	1	1	**1**
Liu-Shu-zhen	0.8004	0.7381	0.7680	0.8409	0.8004	0.8201	0.9217	0.7940	**0.8531**
Li-Qiang	0.8057	0.6886	0.7426	0.9412	0.7968	**0.8630**	0.8962	0.8208	0.8569
Li-Na	0.9487	0.7719	0.8512	0.9870	0.8865	0.9340	0.9870	0.9870	**0.9870**
Li-Gui-ying	0.8871	0.9124	0.8996	0.9879	0.8938	**0.9385**	0.9778	0.8813	0.9271
Michelle	0.9769	0.7205	0.8293	0.9549	0.8146	0.8792	0.9672	0.9498	**0.9584**
Mary	0.9520	0.6828	0.7953	1	0.9290	**0.9632**	1	0.9001	0.9474
Johnson	0.9620	0.8120	0.8807	0.9573	0.8083	0.8765	0.9593	0.8595	**0.9067**
Wang-Gang	0.8130	0.8171	0.8150	0.7804	0.9326	0.8498	0.8143	0.9185	**0.8633**
Wang-Tao	1	0.9323	0.9650	0.9573	0.9485	0.9529	0.9897	0.9768	**0.9832**
Chen-Zhi-qiang	0.9732	0.8401	0.9017	0.9891	0.9403	0.9641	0.9891	0.9564	**0.9725**
Average	0.9153	0.7916	0.8504	0.9451	0.8864	0.9128	0.9548	0.9131	**0.9323**

Table 2 comparison of feature weight computing method (highest F-measure)

	Optimal			TO-CMS			PK3		
	#pre	**#rec**	**#F**	**#pre**	**#rec**	**#F**	**#pre**	**#rec**	**#F**
Lewis	1	1	1	**1**	**1**	**1**	0.8575	1	0.9233
Liu-Shuzhen	0.9217	0.7940	0.8531	0.8466	0.8433	0.8450	0.5451	0.9503	0.6928
Li-Qiang	0.8962	0.8208	0.8569	**0.8962**	**0.8208**	**0.8569**	0.7897	0.9335	0.8556
Li-Na	0.9870	0.9870	0.9870	**0.9870**	**0.9870**	**0.9870**	0.9870	0.9016	0.9424
Li-Gui-ying	0.9778	0.8813	0.9271	**0.9778**	**0.8813**	**0.9271**	0.8750	0.9427	0.9076
Michelle	0.9672	0.9498	0.9584	0.9482	0.9498	0.9490	**0.9672**	**0.9498**	**0.9584**
Mary	1	0.9001	0.9474	0.8545	0.9410	0.8957	0.8698	0.9410	0.9040
Johnson	0.9593	0.8595	0.9067	0.9524	0.8648	0.9066	0.2423	0.9802	0.3885
Wang-Gang	0.8143	0.9185	0.8633	0.9255	0.7102	0.8036	0.5198	0.9550	0.6732
Wang-Tao	0.9897	0.9768	0.9832	0.8594	0.9767	0.9144	0.9700	0.9768	0.9734
Chen-Zhi-qiang	0.9891	0.9564	0.9725	0.8498	1	0.9188	0.8499	1	0.9188
Average	0.9548	0.9131	0.9323	0.9179	0.9068	0.9095	0.7703	0.9574	0.8307

Table 3 comparison of cluster-stopping measures' performance

name	**Entity**	**Created Labels**
Lewis	Person-1	巴比特(Babbitt),辛克莱·刘易斯(Sinclair Lewis),阿罗史密斯(Arrow smith),文学奖(Literature Prize),德莱赛(Dresser),豪威尔斯(Howells),瑞典文学院(Swedish Academy),舍伍德·安德森(Sherwood Anderson),埃尔默·甘特利(Elmer Gan Hartley),大街(street),受奖(award),美国文学艺术协会(American Literature and Arts Association)
	Person-2	美国银行(Bank of America),美洲银行(Bank of America),银行(bank),投资者(investors),信用卡(credit card),中行(Bank of China),花旗(Citibank),并购(mergers and acquisitions),建行(Construction Bank),执行官(executive officer),银行业(banking),股价(stock),肯·刘易斯(Ken Lewis)
	Person-3	单曲(Single),丽昂娜(Liana),专辑(album),丽安娜(Liana),丽安娜·刘易斯(Liana Lewis),利昂娜(Liana),空降(airborne),销量(sales),音乐奖(Music Awards),玛丽亚·凯莉(Maria Kelly),榜(List),处子(debut)、
	Person-4	卡尔·刘易斯(Carl Lewis),跳远(long jump),卡尔(Carl),欧文斯(Owens),田径(track and field),伯勒尔(Burrell),美国奥委会(the U.S. Olympic Committee),短跑(sprint),泰勒兹(Taylors),贝尔格莱德(Belgrade),维德·埃克森(Verde Exxon),埃克森(Exxon)

	Person-5	泰森(Tyson),拳王(King of Boxer),击倒(knock down),重量级(heavyweight),唐金(Don King),拳击(boxing),腰带(belt),拳手(Boxing),拳(fist),回合(bout),拳台(Ring),WBC
	Person-6	丹尼尔(Daniel),戴·刘易斯(Day Lewis),血色(Blood),丹尼尔·戴·刘易斯(Daniel Day Lewis),黑金(There Will Be Blood),左脚(left crus),影帝(movie king),纽约影评人协会(New York Film Critics Circles),小金人(the Gold Oscar statues),主角奖(Best Actor in a Leading Role),奥斯卡(Oscar),未血绸缪(There Will Be Blood)

Table 4 Labels for "Lewis" clusters

On the basis of text clustering result that obtained from the Trade-off based cluster-stopping measure experiment in Table 3, we try our labelling method mentioned in subsection 4.3. For each cluster, we choose 12 words with highest score as its label. The experiment result demonstrates that the created label is able to represent the category. Take name "刘易斯 Liu-Yi-si 'Lewis'" for example, the labeling result shown as Table 4.

5.3 Discussion

From the test result in table 2, we find that our feature weight computing method can improve the Chinese personal name clustering disambiguation performance effectively. For each personal name in test dataset, the performance is improved obviously. The average value of optimal F-measures for eleven names rises from 85.04% to 91.28% by using the whole dataset *D* for calculated *idf*, and rises from 91.28% to 93.23% by using *mi_weight*. Therefore, in the application of Chinese text clustering with constraints, we can compute pointwise mutual information between constraints and feature, and it can be merged with feature weight value to improve the clustering performance.

We can see from table 3 that trade-off point based cluster-stopping measure (*TO_CSM*) performs much better than *PK3*. According to the experimental results, *PK3* measure is not that robust. The optimal number of clusters can be determined for certain data. However, we found that it did not apply to all cases. For example, it obtains the optimal estimation result for data "Michelle", as for "Liu Shuzhen", "Wang Gang" and "Johnson", the results are extremely bad. The better result is achieved by using *TO_CSM* measure, and the selected results are closer to the optimal value. The *PK3* measure uses the mean and the standard deviation to deduce, and its processes are more complicated than *TO_CSM*'s.

Our cluster labeling method computes the features' score with formula (10). From the labeling results sample shown in Table 4, we can see that all of the labels are representative. Most of them are person and organizations' name, and the rest are key compound words. Therefore, when the clustering performance is good, the quality of cluster labels created by our method is also good.

6 Future Work

This paper developed a clustering algorithm of multi-document personal name disambiguation, and put forward a novel feature weight computing method for vector space model. This method computes weight with the pointwise mutual information between the personal name and feature. We also study a hybrid criterion function based on trade-off point and put forward the trade-off point cluster-stopping measure. At last, we experiment on our score computing method for cluster labeling.

Unsupervised personal name disambiguation techniques can be extended to address the problem of unsupervised Entity Resolution and unsupervised word sense discrimination. We will attempt to apply the feature weight computing method to these fields.

One of the main directions of our future work will be how to improve the performance of personal name disambiguation. Computing weight based on a window around names may be helpful. Moreover, word-based text features haven't solved two difficult problems of natural language problems: Synonym and Polysemy, which seriously affect the precision and efficiency of clustering algorithms. Text representation based on concept and topic may solve the problem.

Acknowledgments
This research is supported by National Natural Science Foundation of Chinese (No.60675035) and Beijing Natural Science Foundation (No.4072012)

References

Al-Kamha. R. and D. W. Embley. 2004. Grouping search-engine returned citations for person-name queries. In *Proceedings of WIDM'04*, 96-103, Washington, DC, USA.

Bagga and B. Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of 17th International Conference on Computational Linguistics*, 79–85.

Bagga, Amit and B. Baldwin. 1998. A*lgorithms for scoring co-reference chains.* In Proceedings of the First International Conference on Language Resources and Evaluation Workshop on Linguistic co-reference.

Chen Ying and James Martin. 2007. Towards Robust Unsupervised Personal Name Disambiguation, *EMNLP 2007*.

Chen Lifei, Jiang Qingshan, and Wang Shengrui. 2008. A Hierarchical Method for Determining the Number of Clusters. *Journal of Software*, 19(1). [in Chinese]

Chung Heong Gooi and James Allan. 2004. Cross-document co-reference on a large scale corpus. In S. Dumais, D. Marcu, and S. Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, 9–16, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

Gao Huixian. *Applied Multivariate Statistical Analysis*. Peking Univ. Press. 2004.

G. Salton and C. Buckley. 1988. *Term-weighting approaches in automatic text retrieval*. Information Processing and Management,

Kulkarni Anagha and Ted Pedersen. 2006. How Many Different "John Smiths", and Who are They? In *Proceedings of the Student Abstract and Poster Session of the 21st National Conference on Artificial Intelligence, Boston, Massachusetts.*

Mann G. and D. Yarowsky. 2003. Unsupervised personal name disambiguation. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, 33–40, Edmonton, Canada.

Niu Cheng, Wei Li, and Rohini K. Srihari. 2004. Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction. In *Proceedings of ACL 2004*.

Ono. Shingo, Issei Sato, Minoru Yoshida, and Hiroshi Nakagawa2. 2008. Person Name Disambiguation in Web Pages Using Social Network, Compound Words and Latent Topics. T. Washio et al. (Eds.): *PAKDD 2008, LNAI 5012*, 260–271.

Song Yang, Jian Huang, Isaac G. Councill, Jia Li, and C. Lee Giles. 2007. Efficient Topic-based Unsupervised Name Disambiguation. *JCDL'07*, June 18–23, 2007, Vancouver, British Columbia, Canada.

Ted Pedersen and Kulkarni Anagha. 2006. Automatic Cluster Stopping with Criterion Functions and the Gap Statistic. In *Proceedings of the Demonstration Session of the Human Language Technology Conference and the Sixth Annual Meeting of the North American Chapter of the Association for Computational Linguistic*, New York City, NY.