社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－DBS－140 (II) (34)
2006／7／13

知識ベースを用いた人名検索時の曖昧性の解消

ヴークァン ミン† 正田 備也†† 高須 淳宏†† 安達 淳††

† 東京大学情報理工学系研究科 〒113-8656 東京都文京区本郷 7-3-1
†† 国立情報学研究所 〒101-8430 東京都千代田区一ツ橋 2-1-2
E-mail: †{vuminh,masada,takasu,adachi}@nii.ac.jp

あらまし 人名で検索するとき，同姓同名のため，検索結果に複数の人に関する文書が含まれることが通例である．検索結果をそれぞれの人に関する文書クラスタに分ける手法について検討した．文書間の類似度を計り，同じ人に関する文書かどうかを推測する必要があるが，先行研究では，ベクトル空間モデル法や固有名詞抽出法に基づいて文書間の類似度を計っている．我々は知識ベースを用いて，文書間の共通コンテキストを見つけて，共通コンテキストの重みを計り，文書間の類似度を測定する手法を提案する．実験により，我々の提案手法が先行手法より優れていると確認された．
キーワード 人名検索，曖昧解消，文書類似度

Name Disambiguation in Web Search Using Knowledge Base

Quang MINH VU†, Tomonari MASADA††, Atsuhiro TAKASU††, and Jun ADACHI††

† Graduate School of Information Science and Technology, The University of Tokyo Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-8656 Japan
†† National Institute of Informatics Hitotsubashi 2-1-2, Chiyoda-ku, Tokyo, 101-8430 Japan
E-mail: †{vuminh,masada,takasu,adachi}@nii.ac.jp

Abstract Results of queries by personal names often contain documents related to several people because of namesake problem. In order to discriminate documents related to different people, it is required an effective method to measure document similarities and to find out relevant documents of the same person. Some previous researches have used cosine similarity method or have tried to extract common named entities for measuring similarities. We propose a new method which uses web directories as knowledge base to find out shared contexts in document pairs and uses the measurement of shared contexts as similarities between document pairs. Experimental results show that our proposed method outperforms cosine similarity method and common named entities method.
Key words Personal name searching, name disambiguation, document similarity

1. Introduction

The prevalence of internet in daily life has made the World Wide Web(WWW) space become a huge resource of information. Information in the WWW come from many sources, from websites of companies, organizations, from websites groups of people, from personal homepages of people, etc. Such an explosive information environment of information contains many knowledge, but most of these enormous information are hidden to end-users. For most end-users only a very little amount from the WWW meet their information needs and bring them added value. In order to extract such little valuable information, it is very important to create effective methods to mine from the WWW useful information that end-users are interested in.

Searching engines are developed to help end-users in finding their interested information. End-users send to search engines queries containing important terms to express their information needs. Searching engines search in their databases documents that are related to query terms and present them to end-users. Search engines sort result documents in the order of documents' relationship to queries. However, as most end-users only read some top documents, ranking of documents prevents end-users from retrieving useful documents if they stay far behind from the top. When end-users search for information that only exists scarcely in the WWW, that information tends to hide deeply in the result set.

In order to improve the convenience and the usefulness of searching engines, some researches [1], [2] are trying to improve the method of presentation of result documents to end-users. Two new methods are proposed and researchers are improving them in many applications. One method is a method of clustering retrieved results, another is a method of interactive searching method. The method of clustering retrieved results divides results into groups of documents about the same topic and shows

these clusters to end-users. End-users select clusters closing to their information needs and search more scrutinizingly in those clusters. The method of interactive searching instead select some keywords from result documents and present them to end-users. End-users select appropriate keywords and redo searching using new keywords in their queries.

Our research is trying to cluster retrieved results into groups and especially we focus on clustering results of queries about people. When end-users use queries containing personal name to find information about a certain person, results from searching engines often contain documents related to several people because of namesake problem. Therefore we attempt to divide results into groups, each group only contains documents related to one person. In order to discriminate documents of several people some previous researches have tried to extract contexts of people in results and use them for discrimination [3], or have utilized hyperlink information [4] or have utilized named entities [5] in documents to find related documents. In our research we try to measure similarities among documents, and use these similarities for discrimination process. We propose a new method to measure similarities among documents and compare it with some other traditional similarity measuring methods. Then we apply this method into the problem of name disambiguation in web search.

The rest of this paper is organized as follows. In section 2., we give the problem statements of our research. In section 3., a summarization of related researches is given. Then in section 4., we propose a new method to measure similarities among documents. We will present our new idea for measuring similarity and give detail of calculation process to realize that idea. Experiment results and comparison with traditional similarity measuring method are given in section 5.. Discussion of merit and demerit of our proposed method is mention in section 6.. Finally section 7. will give the conclusion and state our future work to improve effectiveness and feasibility of our method.

2. Problem statements

In this section we will summarize the problem that we are trying to solve. We will state input data to our system, output results of our system to end-users and sketch the operation in our system.

My research objective is to help end-users to exploit knowledge data from the web more easily. The input for our system are searching results of personal name queries from searching engine. Our system try to cluster searching results of personal name queries into groups so that each group only contains documents related to one person. These groups of documents are output of our system to end-users. We think that showing searching results in separate groups of people will help users to navigate throughout result sets more effectively.

Our system has two main steps to discriminate documents. In the first step, it try to measure similarities between every document pairs. We will propose a new method of measuring document pair similarities that can be applied to web documents easily. In the second step, our system use the document similarities measured in step 1 to try to cluster documents into groups. In this paper, we use a very simple clustering algorithm. We select document pairs whose similarities are larger than a threshold and draw a graph using selected pairs as edges. A connect part of this graph forms a group of documents. These groups are output results of our discrimination system.

3. Related researches

In this section we will summarize previous researches of name disambiguation applied in some circumstances and applications: name disambiguation in newspaper articles, name disambiguation in communities and name disambiguation in the web.

Name disambiguation in newspaper articles is among early researches of this kind. As the same person tend to appear in a series of articles under the same context, some methods that are strong at measuring weight of sharing context between documents are used. the vector space model [9] have been applied for this problem [3], [11], [12]. In [11], Amit Bagga et al. used vector space model and tf_idf term weight to measure sharing context weight. In [12], Chung Heong Gooi et al. used Kullback-Leiber Divergence [10] method to measure distance of two distributions of terms in two documents and used this distance for measuring sharing context. In [3], Ted Pedersen et al. used log-likelihood to measure weight of co-occurrence of word pairs. A word pair represents relationship between two words. Each term is represented by a vector constituted by its relationship with other words. A document is presented by an average vector of its all term vectors. Name disambiguation is done by clustering these document vectors.

Name disambiguation was researched for people in a community. In [13], name disambiguation was done on the Internet Movie DataBase (IMDB). In this research, authors use relationships between personal name for disambiguation. Two personal names are considered related if they collocate together. This kind of relationships between personal names are used to construct a graph to represent relationships between personal names. Then this relational graph is used to disambiguate people. The shortcoming of this method is that it strongly depends on relationship between people in a community. In applications other than community, personal relationship is difficult to extract.

As the web has been a popular media to convey information, name disambiguation for web documents have been attracting many researchers. Many of them assumed to take results from output of a searching engine and tried to discriminate people in this document collection. For web documents, the methods that try to find sharing contexts among documents do not work well because of two reasons. In contrast to newspaper article, in the web, many people appear in different places in different topics and context. Therefore just finding documents of the

same context is insufficient to solve the problem. Also a personal document may also contain many contexts, that induce confusion to sharing context evaluation system.

Some approaches have been proposed for the problem of name disambiguation in the web: extraction of keywords to extract sharing context; utilize profile of people to find information that can identify a person; utilize hyperlink information to find condensed connected parts.

In [3], the keyword extraction method try to extract keywords from a set of documents. Extracted important keywords are then used to cluster documents into separate groups. This method requires a large set of documents in order to extract keywords effectively. For example, experiments in this researches have been carried on famous people like Bush, Tony Blair, David Beckham, Zidane. Therefore it works for the cases of not so much famous people, the keyword extraction method is inefficient.

The hyperlink information is an effective information resource in the web [4]. In many cases, people link two pages because they have some common context, so condensed url connected pages seem to form a set of documents on the same topic. However using hyperlink information to cluster web documents face the same problem of scarceness of data as keyword extraction method. When the number of documents is few, condensed url connected pages are very little, making the recognition of common context connected parts becoming more difficult.

Some researches have tried to find personal profile in the web to create profiles of people and use profiles for discrimination tasks. In [14], authors tried to use pattern matching to extract personal information like birthdays, birthplaces, ages and use these information to identify people. Its shortcoming is that it can only be applied for webpages like curriculum vitae pages. In [5], authors tried to use author information of books, census data, list of places, organization as dictionaries to help the extraction of profile. This method can be applied if information of people to be discriminated exists in directories. In [15], authors tried to use natural language processing technique to recognize named entities in documents. This method depends largely on performance of named entity recognition program, which is hard to show performance when processing web documents.

Previous researches work well in some certain circumstances (people in communities with strong relationships between people, very famous people, authors of books). However we need to solve name disambiguation in more general circumstances in order to increase its applicable applications. In the next section, we will introduce a new method that can work well with more general circumstances of people.

4. Similarity via Knowledge Base (SKB)

In this section we propose a new method to measure similarities between document pairs and to apply this measurement method for name disambiguation in web search. First we introduce the idea of our method for measuring similarity between document pairs. Then we present a calculation algorithm to realize this idea. Finally we present a simple clustering algorithm using proposed similarity measuring method for the problem of name disambiguation in web search.

4.1 Introduction of our approach

In information retrieval, the tasks of finding common contexts of documents, judging if documents are related to each other are very crucial in many applications. Many traditional methods such as vector space model method, keyword extraction method, feature extraction from document cooperating with document clustering method, etc are proposed. The essence in these methods is to find common part among documents, to measure weight of common part. In some previous researches on name disambiguation in web search, traditional similarity judging methods like vector space model method and keyword extraction method have been used, but they show some drawbacks.

Vector space model method try to measure the similarity using all terms in documents. It works well in the application of finding co-reference of articles in newspaper. An article in newspaper often discusses only one topic so two articles on a same topic share many common words. However in case of web documents related a person, that person may appear in many context, documents related to him may contain many different contexts, so vector space model method do not work well in this application.

Keyword extraction method can remedy the drawback of vector space model method. This method try to extract keywords related to contexts in web documents and use these keywords in measuring similarity. The performance of keyword extraction is verified when apply to people to whom many documents relate. For a famous person set of documents relate to him is large, keywords appear frequently so keyword extraction is easier. However when the number of documents is moderate or small, keywords appear only few times making the keyword extraction algorithm fails to separate keywords from other words.

Human's ability on recognizing of keywords is extremely strong. Even we read only one document, we can understand its topic, separate keywords of the topic from other words. For example when we find words like *"algorithm"*, *"programming"*, etc in an article, we understand that the article is talking about computer and we can find out other computer related words in the article. We think that human can do that because human use other knowledge outside document when reading it. Besides document that we are reading, we use our knowledge to understand it, and we use the knowledge on computer to recognize that it is a computer related document.

We try to imitate human's method of recognizing keywords for the problem of finding documents related to the same person in web search. To do that, we propose a method that use a prepared knowledge base to assist the task of similarity measurement. We name this method as **Similarity via Knowledge Base (SKB)**. The details is as follows. We prepare a knowledge base containing many directories, each directory is a collection of docu-

ments on a same topic. This knowledge base of directories plays the same role as human's knowledge. We use these directories to help the task of calculating document similarities as follow. First we find from knowledge base some directories that have close topic with a document. Common words between directory and document are extracted. Then we use two sets of extracted common words from two documents for the calculation of similarity between two documents. The detail of calculation algorithm is explained in the next sub section.

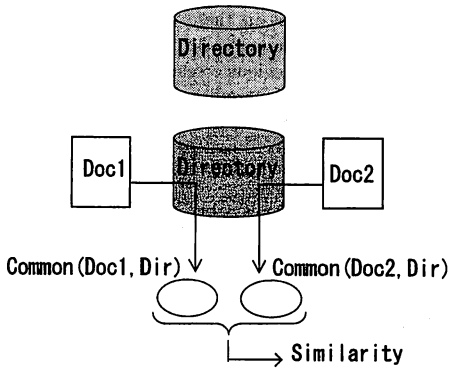

Figure 1 Similarity via Knowledge Base

4.2 Calculation algorithm

Calculation algorithm of similarity using knowledge base has three steps as follows.

(1) Preprocessing
(2) Similarity between a directory and a document
(3) Similarity between document pairs using knowledge base

The following sub sections will give details of each step.

4.2.1 Preprocessing

Preprocessing step is to remove unrelated noisy from documents. We first remove stop words (words appear in many documents and do not contain much specific information, e.g.: *a, the, be, etc*). Then we do stemming to group different forms (noun single forms, noun plural forms, gerund forms, etc) of the same term. Finally as a general webpage usually contains information of several people only surrounding part of a personal name contains related information. Therefore we use only 50 terms in front and 50 terms behind personal name to create a bag of words representing that people.

4.2.2 Similarity between a directory and a document

In this step, we attempt to find directories that are closed to a document and find important terms in directories to represent basic context of document.

We render a document d through a directory Dir as follow.

$$idf(t, TREC) = log(\frac{N}{df}) \quad (1)$$

$$tf\_idf(t, d) = tf(t, d) \times idf(t, TREC) \quad (2)$$

$$tf\_idf(t, Dir) = sum\_tf(t, Dir) \times idf(t, TREC) \quad (3)$$

$$weight(t, d, Dir) = \sqrt{\frac{tf\_idf(t, d) \times tf\_idf(t, Dir)}{num\_term(Dir)}} \quad (4)$$

Document frequencies of a term in the TREC-Web collection [19] to get representativeness of a term in the universal corpus. In equation 4, we divide by $num\_term(Dir)$ to normalize the similarities in order to make them comparable among directories.

Calculate $weight(t, d, Dir)$ for each term in doc d, then select and take the sum of top 10 terms with highest score $weight(t, d, Dir)$.

Represent the set of top 10 highest score words as $Render(d, Dir)$.

Represent the sum as $doc\_dir\_sim(d, Dir)$.

For each document we keep top 10 directories $Dir_1, Dir_2, ...Dir_{10}$ of highest score $doc\_dir\_sim(d, Dir)$ (thereafter we call these directories as rendering directories of document d).

4.2.3 Similarity between document pairs using knowledge base

In this step we try to measure common terms in two bags of words in two documents.

Let a pair of documents to be measured as (d_1, d_2). Rendering directories of d_1, d_2 are $Dir_1, Dir_2, ...Dir_{10}$ and $Dir'_1, Dir'_2, ...Dir'_{10}$ respectively. For each Dir in $Dir_1, Dir_2, ...Dir_{10}, Dir'_1, Dir'_2, ...Dir'_{10}$ we calculate the similarity of d_1, d_2 via Dir as follow:

$$contribute(t, d_1, d_2, Dir) = weight(t, d_1, Dir) \times weight(t, d_2, Dir) \quad (5)$$

$$SIM(d_1, d_2, Dir) = \sum_t contribute(t, d1, d2, Dir) \quad (6)$$

where $t \in Render(d_1, Dir) \cap Render(d_2, Dir)$.

Then the similarity between pair (d_1, d_2) is calculated as

$$SIM(d_1, d_2) = \max_{i=1..n} SIM(d_1, d_2, Dir_i) \quad (7)$$

4.3 Clustering algorithm

In order to separate search result documents into groups, each group only contains documents related to the same person, we need a clustering algorithm using measured similarities. We use a very simple algorithm of clustering. Using training data, we manually tune an appropriate similarity threshold. We create a graph whose nodes are documents and whose edges are connections between document pairs with similarities larger than the tuned threshold. The constructed graph will contain several separated connected parts. We will assume that each part only contains document of the same person and show these groups results to end-users.

5. Experiment

In this section we carry experiments of our proposed SKB method on some test data sets. In order to verify the improvement of our SKB method, we also do the experiments on the same data sets using three other baseline methods and compare our SKB method with these baseline methods. First we describe the three baseline methods. Then we give details about data sets used in the experiments. Finally we report the experiment results of each method and compare the results of our method to that of other methods.

5.1 Baseline methods

We choose three methods as baseline method to compare with our method. They are Vector Space Model method, Support Vector Machine, Named Entities Recognition method. The following subsections will give details about each method.

5.1.1 Vector Space Model method

In Vector Space Method (VSM) method, we do preprocessing same as preprocessing in our SKB method. That is we select 50 non-stop words before and 50 non-stop words after each personal query name. Using this bag of words we construct a document vector whose constituents are tf_idf values of words in the bag. We use inner vector product of document vectors as similarity measurement of document pairs.

5.1.2 Support Vector Machine method

As our SKB method uses knowledge base as assistance information, we compare our method with another traditional method that can utilize information from knowledge base. We choose Support Vector Machine (SVM) method because SVM can use directories in knowledge base to train a classifier. The details process is as follows.

The number of directories chosen for knowledge base is 56 (we will mention the details of these directories in the next sub section). We consider these 56 directories as 56 topics and attempt to classify documents under discriminated to these 56 directories. A classifier is build based on Support Vector Machine method using 56 directories as its training data set. Then for each document, the classifier calculate 56 decision values that current document belong to 56 topics. We build a vector to represent document using these 56 decision values. Then the inner vector product of a document pair is used for similarity measurement of that pair.

5.1.3 Named Entities Recognition method

In [5], authors use Named Entities Recognition (NER) method for the measurement of document similarities. We use the [6] NER tool to extract named entities inside document and build a document vector using these named entities. Constituents of vector are binary value (1 if a named entity appear in the document, otherwise 0). The inner vector product between document vectors is used for similarity measurement.

5.2 Data sets

5.2.1 Knowledge base directories

We choose directories in dmoz.org [7] for knowledge base directories. We choose 56 specific directories from various general topics like: art, business, computer, games, history, home, news, recreation, science, shopping, society and sports. Each directory contain about 40 ~ 50 documents.

5.2.2 Test sets

We get results from the Google search engine [8] for 6 queries for 6 names of 6 people as shown in the following table.

Name	Research field	#Total doc	#Related doc
Adachi Jun	Information retrieval	71	23
Tom M. Mitchell	Machine learning	73	34
J. M. Roberts	History	89	49
Christopher Manning	Natural language processing	78	50
Sakai Shuichi	Computer architecture	83	44
Tanaka Katsumi	Database, knowledge base system	85	49

Using these 6 document sets, we carry 6 experiments to separate documents related to our chosen people from other noise documents.

We also do two experiments trying to separate documents related to several people at the same time. We mix 3 document sets of 3 people and try to divide into 4 groups: 3 groups contain related documents to 3 people and 1 group contains other noise documents. The two sets of several people are as follows.

(1) "Adachi Jun" vs. "Tom M. Mitchell" vs. "J. M. Roberts" vs. others

(2) "Adachi Jun" vs. "Sakai Shuichi" vs. "Tanaka Katsumi" vs. other

In the first experiment, 3 people have different research field while in the second experiment, 3 people have quite close research field.

5.3 Evaluation method

We use F measure method to evaluate performance of each method.

Let S_{ans} and S_{res} be the set of correct answer documents and the set of documents retrieved by the system respectively. Then the calculation of $F_{measure}$ is as follows.

$$Precision(P) = \frac{|S_{res} \cap S_{ans}|}{|S_{res}|}$$

$$Recall(R) = \frac{|S_{res} \cap S_{ans}|}{|S_{ans}|}$$

$$F_{measure} = \frac{2P \times R}{P + R}$$

For the evaluation of experiments discrimination of document related to one person from other documents, the way of using F measure is straight forward as we have only one correct answer set and one result set.

For the evaluation of experiments discrimination of document related to several people, the way of using F measure for evaluation is as follows.

Let the sets of correct answer documents for 3 people be $S_{ans1}, S_{ans2}, S_{ans3}$.

Let the sets of result documents answer returned by the system as $S_{res1}, S_{res2}, S_{res3}, ...S_{resN}$.
(We notice that the system do not know the number of people in the correct answer so the number of groups (N) returned by the system may differs 3).

To evaluate returned results, we find three separated pairs $(S_{ans1}, S_{res\_i}), (S_{ans2}, S_{res\_j}), (S_{ans3}, S_{res\_k})$, whose $F_{measure}(S_{ans}, S_{res})$ values are top three among all possible (S_{ans}, S_{res}) pairs. The average of the selected top three $F_{measure}$ is used to evaluate performance of the system.

5.4 Experimental result

We have carried out two experiments. In the first experiment we compare the performance between four methods: Vector Space Model (VSM), Support Vector Machine (SVM), Named Entities Recognition (NER) and Similarity via Knowledge Base (SKB). For SVM and SKB methods, we use 56 directories mentioned above. In the second experiment, we test the performance of SVM and SKB methods when the structure and characteristics of knowledge base directories vary. We use two sets of knowledge base directories, one is above mentioned 56 directories, another is the set of three directories: "Text Mining", "Machine Learning" and "History Education". The purpose of second experiment is to verify the stability of SKB methods against structure and characteristic of directories.

Test set	VSM	SVM	NER	SKB
Adachi_Jun	0.50	**0.65**	0.64	0.62
Tom_M_Mitchell	0.69	0.76	0.62	**0.94**
J_M_Roberts	0.73	0.75	0.75	**0.82**
AJ_TMM_JMR	0.21	0.16	0.24	**0.48**
Average	0.53	0.58	0.56	**0.71**

Table 1 Comparison of performance

Figure 2, 3, 4 shows some typical experiment results in the experiment 1. Table 1 shows the best $F_{measure}$ value in each when we vary the similarity threshold.

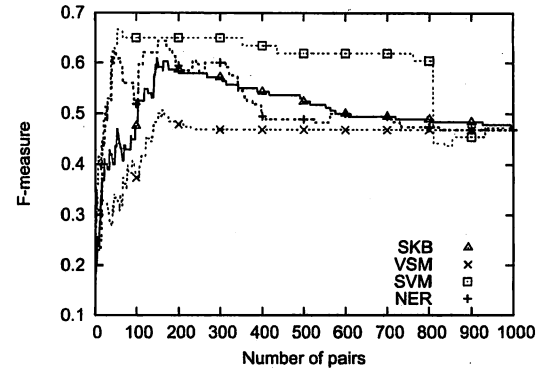

Figure 2 Adachi Jun

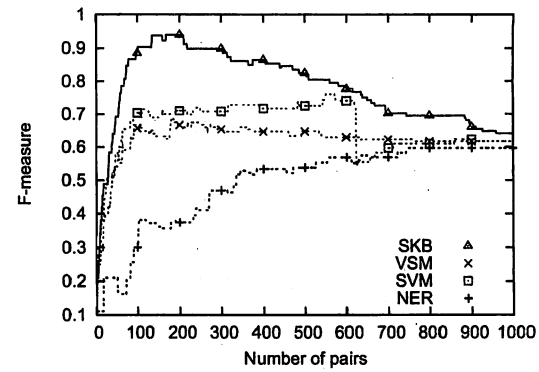

Figure 3 Tom M. Mitchell

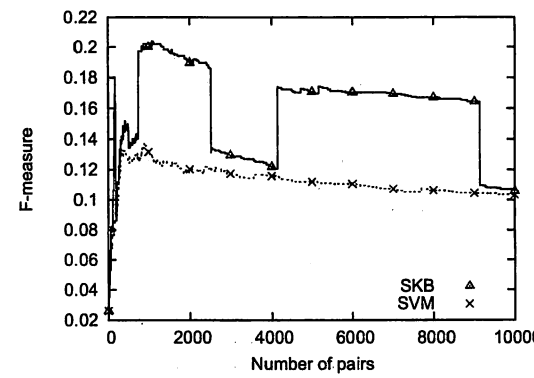

Figure 4 SKB vs. SVM (AJ_SS_TK data set)

6. Discussion

6.1 Comparison

From the figure 2, 3, 4 and table 1, we see that SKB

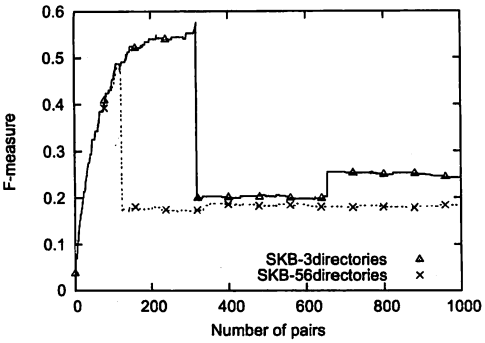

Figure 5 SKB 56 dirs vs. SKB 3 dirs (AJ_TMM_JMR)

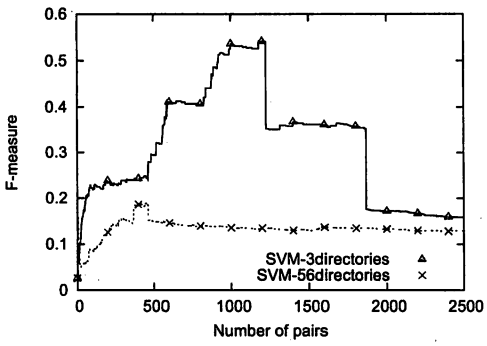

Figure 6 SVM 56 dirs vs. SVM 3 dirs (AJ_TMM_JMR)

outperform other methods (VSM, SVM, NER).

VSM shows worse performance than SVM method and SKB method. This is because in VSM, co-occurrence of noise term (terms are not strongly related with topics) may affect the precision of measurement of similarities.

NER works well in some cases but in some others cases, it shows very poor performance. This is because NER method strongly depends on output results of NER tools.

In the experiments of separation documents related to one person from other documents SVM method achieves comparable performance as our SKB method. However in the experiment of separation documents related to 3 people the difference between SKB and SVM become clear. This is because SVM method only consider how close a document is to a directory while in our SKB method not only the closeness can be measured, but also the important keywords related to a topic can be extracted. With the ability of extracting important keywords, our SKB method can discriminate people in near topics if each person has different topic related keywords. The performance of SKB over the mixture of 3 data sets of 3 people in near topic (Adachi Jun, Sakai Shuichi, Tanaka Katsumi) verify this ability of our SKB method.

Our SKB method is also better than SVM method because it depends less on directories structure. As can be seen from the second experiment, when directories' topics are disjoint among each other (experiment with 3 directories), SVM shows performance as well as our SKB method. However when some topics of directories overlap among other, performance of SVM decline dramatically while SKB still shows reasonable performance. We argue that it is because SVM failed to extract features for classification process. Consider the case when some directories overlap each other (for example: *"Computer.AI"*, *"Computer.Hardware"*, *"Computer.Software"*), while they are disjoint with other directories (for example: *"Sports"*, *"History"*). In such a case features like *"computer"*, *"information"*, *"program"*, *etc* are good features to separate *"Computer.*"* vs. *"Sports"*, *"History"* but they are not good features to separate among *"Computer.AI"*, *"Computer.Hardware"*, *"Computer.Software"*. Therefore SVM system are confused when selecting features for classification process. On the other hand, SKB method do not try to classify a document into a directory. Instead it only uses directory to assist the extraction of keywords so the overlap among directories do not affect the assistance very much.

6.2 Merit demerit of our SKB method

6.2.1 Merit points

Our SKB method has merit points as well as demerit points. Its most impressive merit point is the ability of extracting keywords by topics. This merit causes two improvement in measuring similarity among documents. First, it can focus similarity measurement on important keywords and reduce noise induced by topic unrelated words. Second, it helps us to re-evaluate the importance of topic related keywords more precisely. As we can see traditional tf_idf method of evaluate terms' importance only consider the term frequencies in a general corpus. On the other hand in our SKB method as topic related term appear more frequently in topic related directory than in the general corpus our evaluation method can improve the measurement of terms' importance.

As our SKB method can extract topic related keywords, it can help end-users to navigate the retrieved result more easily by showing these keywords to end-users. These keywords will help end-users to grasp the overall content in each group of documents.

Our SKB method can be extend to apply in other applications that require the task of name disambiguation like co-citation application or discriminate people in blog sites. To apply our SKB method into a new application in a new domain, we have to prepare directories that reflect different contexts appearing in that domain.

Our SKB method can be used for larger number of people in the web. Some other previous methods (e.g. keyword extraction based method, hyperlink information based method) work well for very famous people but are not appropriate for people whose related documents are not so many. In contrast, with SKB method, even the number of related documents is small, with the assistance of knowledge base directories, the extraction of keywords and topic words is easier.

6.2.2 Demerit points

Our SKB method also has some demerit points. First, as it needs a knowledge base directory system as assistance information, the structure of this directory system is very crucial. In experiments report in this paper, we use 56 specific directories. In future we have to examine the performance with more generic directories and with the more number of specific directories. Computation cost is the second demerit of our system. Comparing to other methods our SKB method has to calculate similarity between a document and every directory. We have to find method to reduce computation cost in order to make our method become more feasibility.

6.3 Future works

We are going to continue research on our SKB method so that it can be applied in real application. Some challenges are as follows. The first challenge is to decide a similarity threshold for the selection of strong related document pairs. The second challenge is to improve clustering method. At this moment, we use a very simple clustering method: a connected graph forms a group of document. From the experiment results, we see that the performance decrease when there exists a bridge in the connected part (a bridge is an edge that when remove it the connected part divides into two parts). Grouping only strong connected parts may help to reduce this error. Furthermore, we need to treat document similarities more sophisticatedly in order to improve clustering performance.The third challenge is to construct component of directories so that SKB can utilize it easily. We are going to organize directories hierarchical structure, specific directories are children of generic directories. Using this hierarchical structure directories, we hope to reduce computation cost of our system.

7. Conclusion

In this research we focus on the problem of clustering searching results into groups so that end-users can easily navigate through the results and find out his looking information. We specially focus on clustering searching results of personal name queries. We have proposed a new method to measure similarities between documents: Similarity via Knowledge Base (SKB), we use knowledge base to help to find out topic words, important keywords in documents and to calculate sharing terms among documents. Our method outperform other traditional method of measuring similarities in terms of better extraction of topic keywords, separate sharing topic terms from sharing noise terms. Our method also outperform some previous discrimination system in term of it can be used to discriminate documents of people whose number of related documents are small. In this research, SKB method is used to discriminate searching results but it can be also applied for other applications that require discrimination like cocitation problems, discriminate people in blog sites, etc. For each application, we need to prepare an appropriate knowledge base directories for that application.

References

[1] Oren Zamir, Oren Etzioni. Grouper: A Dynamic Clustering Interface to Web Search Results. The Eighth International World Wide Web Conference.

[2] Hitoyuki Toda, Ryoji Kataoka. A Search Result Clustering Method using Informatively Named Entities. ACM Fourteenth Conference on Information and Knowledge Management, CIKM2005.

[3] Ted Pedersen et al.. Name Discrimination by Clustering Similar Contexts. Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics, 2005.

[4] Ron Bekkerman, Andrew McCallum. Disambiguating Web Apprearances of People in a Social Network. The Fourteenth International World Wide Web Conference, WWW2005.

[5] R. Guha, A. Garg. Disambiguating People in Search. The Thirteenth International World Wide Web Conference, WWW2004.

[6] http://www.alias-i.com/lingpipe/

[7] http://www.dmoz.org/

[8] http://www.google.com/

[9] Ricardo Baeza-Yates, Berthier Ribeiro-Neto. Modern Information Retrieval. Addison Wesley Longman Publishing 1999.

[10] Christopher D. Manning, Hinrich Schutze. Foundations of statistical natural language processing. The MIT Press 2003.

[11] Amit Bagga, Breck Baldwin. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. ACL 1998.

[12] Chung Heong Gooi, James Allan. Cross-Document Coreference on a Large Scale Corpus. Technical report, University of Massachusettes, 2004.

[13] Bradley Malin. Unsupervised Name Disambiguation via Social Network Similarity. SIAM ICDM 2005.

[14] Gideon S. Mann, David Yarowsky. Unsupervised Personal Name Disambiguation. Computational Natural Language Learning 2003.

[15] 小野真吾, 吉田稔, 中川裕志. Web における名寄せシステム. 言語処理全国大会 NLP2006.

[16] 白砂健一, 小山聡, 田島敬史, 田中克己. Web の構造情報とプロファイル抽出を用いたオブジェクト識別. DEWS2006.

[17] 木村塁, 戸田浩之, 田中克己. 検索結果スニペットのクラスタリングによる同姓同名人物の特定. DEWS2006.

[18] Xiaojun Wan, Jianfeng Gao, Mu Li, Binggong Ding. Person Resolution in Person Search Results: WebHawk. CIKM 05.

[19] http://trec.nist.gov/