Personal Name Disambiguation in Web Search Using Knowledge Base

Quang Minh Vu ♥ Tomonari Masada ♦
Atsuhiro Takasu ♠ Jun Adachi ♣

Results of queries by personal names often contain documents related to several people because of the namesake problem. In order to differentiate documents related to different people, an effective method is needed to measure document similarities and to find documents related to the same person. Some previous researchers have used the vector space model or have tried to extract common named entities for measuring similarities. We propose a new method that uses Web directories as a knowledge base to find shared contexts in document pairs and uses the measurement of shared contexts to determine similarities between document pairs. Experimental results show that our proposed method outperforms the vector space model method and the named entity recognition method.

1. Introduction

The prevalence of the Internet in daily life has made the World Wide Web(WWW) a huge resource for information. Information in the WWW comes from many sources, including websites of companies, organizations, communities, personal homepages, etc. In such a heterogeneous environment, information about one person tends to be scattered in various places. Suppose we want to search for information about a person. We may send a query containing his name to a search engine and get a set of documents containing his name. However, because of the namesake problem the set of documents may contain documents related to several people. For example, the top 100 pages from the Google search engine for the query *"Jim Clark"* contain at least eight different *Jim Clarks*. Among them, two people with the largest number of pages are *Jim Clark* the Formula one world champion (46 pages) and *Jim Clark* the founder of Netscape (26 pages). It would be more easily for end-users to find their interested person, if we can determine documents related to the same person. For this purpose, to measure the closeness between pairs of documents correctly is very crucial because it directly affects determining performance. In some previous research [1, 2, 3, 9, 10, 11, 12], several methods have been proposed to determine similarities between document pairs. We propose a new method to effectively measure document pair similarities. We use several sets of documents on several topics as intermediate documents to find out shared contexts in document pairs and measure the weight of these shared contexts. These sets of documents can be regarded as a knowledge base, an information source on various topics. We chose to use Web directories for the knowledge base because they are easy to get from the Web. We used the Dmoz Web directories [5] in our research. Our proposed method may be used in cooperation with some already existing methods to improve the disambiguating performance.

♥ Student Member Graduate School of Information Science and Technology, The University of Tokyo vuminh@nii.ac.jp
♦ Regular Member National Institute of Informatics masada@nii.ac.jp
♠ Regular Member National Institute of Informatics takasu@nii.ac.jp
♣ Regular Member National Institute of Informatics adachi@nii.ac.jp

2. Problem statement and related works

Bagga and Baldwin[9] solved the problem of personal name coreference in news articles. They used the vector space model (VSM) [7] to measure similarities between articles. A person appearing in some news articles tends to be related to one event, so that person's relevant documents tend to discuss only one story. However, a person in the Web may appear with more than one event and their relevant documents may be about different topics. Therefore, the VSM model may not work well with web documents as with news articles.

Pederson et al.[1] calculated documents' context vectors using the method *second order context vectors* [17]. Then, they used the documents' context vectors to cluster the documents into groups. However, this approach is suitable only for people whose names appear in a large number of documents because calculation of words's context vectors requires word co-occurrence information from a large number of documents.

Bekkerman and McCallumn[2] proposed a method to extract a group of people simultaneously. People in this group are related to one another so their relevant web pages may share the same topic and be connected. The researches proposed two methods to extract a group of people: one that uses link information in web pages and another that uses the Agglomerative Conglomerative Double Clustering (A/CDC) clustering algorithm to group together web pages with the same topic. The use of this method is limited because when we search for a person on the Internet, we may not know about his social network in advance.

Extraction of personal profiles has been used in some other researches [11, 3, 12]. Mann et al.[11] used the pattern matching method to extract personal profiles (birthday, birth place, occupation, etc). Guha et al.[3] used databases like DBLP [14], Amazon [15] to extract books' author names and research keywords. Wan et al.[12] used natural language processing techniques to extract named entities in documents.

3. Similarity via Knowledge Base (SKB)

3.1 Measuring document similarities

The vector space model (VSM) method measures the weight of terms based on the number of times a term occurs in a document (term frequency) and the number of documents that contain the term (document frequency). It works well when related documents discuss the same specific topic and documents share many common terms. However, a person in the web may appear in different circumstances. Therefore, although his relevant documents may be about the same general topic, their specific topics may be different. In such a case, common terms among documents are very few. When the number of common terms is few, similarities calculated by VSM are not so effective for differentiating documents relevant to different people.

We propose a new method to boost the weight of important terms in order to measure document similarities when the number of common terms is small. Suppose that we have a set of documents that are about topics close to those of a pair of documents. In the pair of documents, because of the small number of documents and the shortness of documents' length, keywords

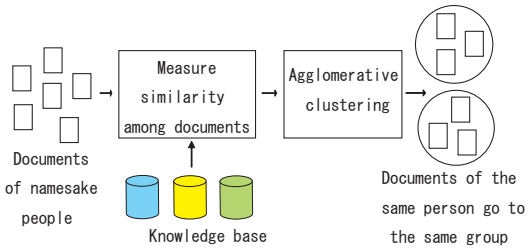

Fig. 1 Name disambiguation system

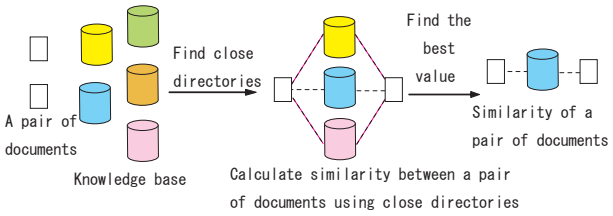

Fig. 2 Measure similarity using a knowledge base

related with the topic may not appear more frequently than other words. However, in the set of documents of a topic close to the document pair, keywords appear more frequently than other words. It is reasonable to assume that keywords in the document pair appear as frequently as they do in the set of documents if the relevant texts are longer. Therefore, we use the frequencies of keywords in the set of documents to modify the frequencies of keywords in the pair of documents.

This approach requires external sets of documents, so we prepared some sets of documents on some topics. We call these sets the knowledge base. Then we used this knowledge base to find document sets that are close in topic to a pair of documents and modify their keywords' term frequencies. We call our method "Similarity via Knowledge Base" (**SKB**). The knowledge base used in our SKB method can be seen as a kind of training data. This kind of offline training data is independent with online people being disambiguated. Also, the preparation of this training data is not expensive because we may use already existing document categories (e.g. web directories) as knowledge base.

3.2 Calculation algorithm

Figure 1 and figure 2 show the overview of the name disambiguation system using the knowledge base. It has three steps as follows.

1. Preprocess documents
2. Find directories from the knowledge base that are close in topic to a document and measure weight of terms using these directories.
3. Measure similarity between a pair of documents using knowledge base.

3.2.1 Preprocessing

We remove stop words and use the Porter algorithm[16] to stem words to their root forms. Since web pages are noisy information source so words being far from a personal name may not relate with the concerned person. Therefore, we only extract terms in a fixed window surrounding the personal name to create a bag of words representing that person. In our experiment, we experimentally set the window size to be 50.

3.2.2 Finding close directories and measuring term weights

The traditional VSM uses the following formulas to calculate term weights.

$$tf\_idf(t,d) = tf(t,d) \times log(\frac{N}{df}) \quad (1)$$

Here $tf(t,d)$ is the frequency of term t in the document d. We use the TREC-Web collection[13] to calculate the inverse document frequency $idf(t, TREC)$ of term t. N is the number of documents in the TREC-Web collection.

Suppose that a directory Dir (a set of documents in the knowledge base) is close in topic to the document d. Then the distribution of a topic's keyword term t in d and Dir should be similar if d is long enough. Therefore, the larger a term t's weight $tf\_idf(t, Dir)$ is, the larger that term t's importance in the document $weight(t,d)$ should be.

We may use $tf\_idf(t, Dir)$ in place of $tf\_idf(t,d)$, but we still want to keep the importance of $tf\_idf(t,d)$, so we choose the geometric mean as follows.

$$weight(t,d,Dir) \propto \sqrt{tf\_idf(t,d) \times tf\_idf(t,Dir)} \quad (2)$$

We have many directories and we want to make this importance comparable among them, so we normalize this importance by dividing it by the size of directories.

$$weight(t,d,Dir) = \sqrt{\frac{tf\_idf(t,d) \times tf\_idf(t,Dir)}{length(Dir)}} \quad (3)$$

We use the following formula to calculate similarity between a document d and a directory Dir.

$$SIM(d,Dir) = \sum_{t \in d \cap Dir} weight(t,d,Dir) \quad (4)$$

Then, for each document d, we select the top k directories $Dir_1, Dir_2, ...Dir_k$ with the highest $SIM(d,Dir)$ values as representative directories for document d.

3.2.3 Measuring document pair similarities

Let (d_1, d_2) denote a pair of documents to be measured. For each Dir_i in the representative directory set of $d1, d2$ we calculate the similarity between (d_1, d_2) via a directory Dir_i as follows

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

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

where $t \in d_1 \cap d_2 \cap Dir$.

Then, we calculate the similarity between $(d1, d2)$ as follows.

$$SIM(d_1,d_2) = \max_i SIM(d_1,d_2,Dir_i) \quad (7)$$

where Dir_i are representative directories of d_1, d_2.

3.3 Clustering documents

Suppose we have two document sets, and each set has only documents related to the same person. If these two document sets are similar enough to each other, both of them may be about the same person, so we merge them together. The similarity between two sets of documents is calculated as follows.

$$SIM(C_1,C_2) = \frac{\sum_{d_i \in C_1, d_j \in C_2} SIM(d_i,d_j)}{|C_1| \times |C_2|} \quad (8)$$

The clustering algorithm is as follows. At the initial step each document itself forms a singleton cluster. Then, we consecutively merge the closest cluster pair until the ratio between the number of clusters and the number of input documents is below a certain threshold.

4. Experiment

4.1 Baseline methods

We chose two methods as baseline methods to compare with our method: the vector space model (VSM) method and the named entity recognition (NER) method.

4.1.1 Vector space model method

In the VSM method, we did preprocessing same as preprocessing in our SKB method: we removed stop words and select 50 words before and 50 words after each personal query name. Using this bag of words, we constructed a document vector whose constituents are $tf\_idf(t,d)$ values of all words in the bag calculated using equation 1. We used the inner vector product of document vectors as the similarity measurement of document pairs.

4.1.2 Named entity recognition method

We used the LingPipe software[4] to extract named entities inside a document and built a document vector using these named entities. Constituents of vector were binary value (1 if a named entity appear in the document, 0 otherwise). The inner vector product between document vectors was used for similarity measurement.

4.2 Data sets

4.2.1 Knowledge base directories

We created a knowledge base by choosing 56 directories in dmoz.org [5]. These directories are on various topics including art, business, computer, games, history, home, news, recreation, science, shopping, society and sports. Each directory contains about 40 to 50 documents.

4.2.2 Test sets

We selected researchers in four fields: computer science, physics, medicine and history. We chose six people from each field, as shown in Table 1. We sent these names as queries to the Google search engine[6] and selected the top 100 document results containing the personal name. Each result set contained documents referring to our selected person and documents referring to other namesakes. After removing the non-html documents, each collection had about 75 to 90 documents, among them about 20 to 60 documents were documents related to our selected person. Hereafter, we call a namesake with 20 to 60 relevant documents in the data set a major person and other namesakes with a few number of relevant documents in the data set minor people.

We tried to create test data as close to those of the real application as possible. In a real information retrieval system, we would not know the number of namesakes that result documents refer to in advance. Also, in the result set, some people would have many relevant documents while other people would have only a few relevant documents. Therefore, we created pseudo-namesake data by mixing two document collections corresponding to the search results for two names of two people in different research fields. This yielded $6 \times 6 \times \binom{4}{2} = 216$ pseudo namesake data. Each pseudo-namesake data created in this way had two major people and several other minor people.

4.3 Evaluation

We evaluated the performance of disambiguating major people as follows.

Table. 1 Data sets

Field	Name
Computer science	Adachi Jun, Sakai Shuichi Tanaka Katsumi, John D. Lafferty Tom M. Mitchell, Andrew McCallum
Physics	Paul G. Hewitt, Edwin F. Taylor Frank Bridge, Kenneth W. Ford Paul W. Zitzewitz, Michael A. Dubson
Medicine	Scott Hammer, Thomas F. Patterson Henry F. Chambers, David C. Hooper Michele L. Pearson, Lindsay E. Nicolle
History	John M. Roberts, David Reynolds Thomas A. Brady, William L. Cleveland Thomas E. Woods, Peter Haugen

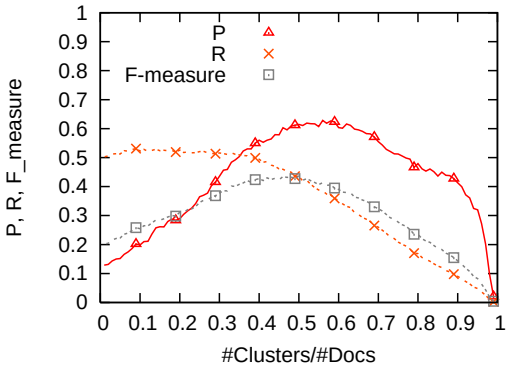

Fig. 3 Performance of VSM method

Labeling documents

We labeled documents in the clusters as follows. We removed clusters whose size was less than or equal to three. From remaining clusters, we selected two clusters, each of the two contained the most number of documents relevant to each major person. We marked documents in each cluster with the label of that major person's name.

Evaluation metrics

Denote $N_{i\_labeled}$ as the number of documents being labeled with ith person's name label ($i = 1, 2$). Denote $N_{i\_correct}$ as the number of documents correctly being labeled with ith person's name label ($i = 1, 2$). Denote N_{i_total} as the total number of documents relevant to ith person ($i = 1, 2$). We calculated the top averaged precision (P_{top_aver}), top averaged recall (R_{top_aver}) of the labeling result. We also calculated the harmonic mean ($F_{measure\_top\_aver}$) of P_{top_aver} and R_{top_aver}.

$$P_i = \frac{N_{i\_correct}}{N_{i\_labeled}} \tag{9}$$

$$R_i = \frac{N_{i\_correct}}{N_{i\_total}} \tag{10}$$

$$P_{top\_aver} = \frac{P_1 + P_2}{2} \tag{11}$$

$$R_{top\_aver} = \frac{R_1 + R_2}{2} \tag{12}$$

$$F_{measure\_top\_aver} = \frac{2P_{top\_aver} \times R_{top\_aver}}{P_{top\_aver} + R_{top\_aver}} \tag{13}$$

4.4 Experimental results

We varied the stopping condition of the clustering algorithm (i.e. the ratio between the number of clusters and the number of input documents) and measured the values $P, R,$ and $F_{measure}$.

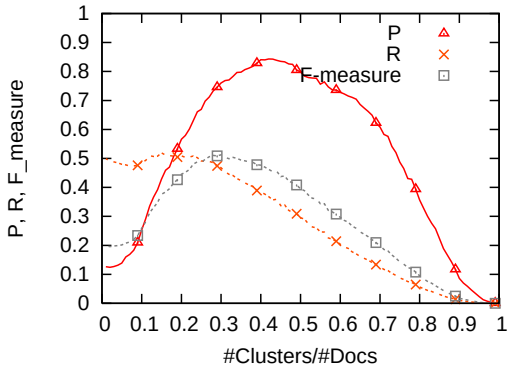

Fig. 4 Performance of NER method

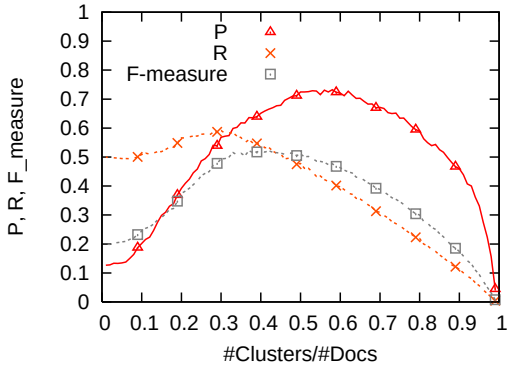

Fig. 5 Performance of SKB method with 56 directories

We carried experiments with 216 pseudo namesake data and took the averaged performance. Figure 3,4,5 show the results for three methods: VSM, NER, and SKB, respectively. As we can see from these figures, in terms of precision, the order of better performance is the NER method, the SKB method and the VSM method. However, in terms of recall, the order becomes the SKB method, the VSM method, and the NER method. In terms of $F_{measure\_top\_aver}$, which considers both the precision and the recall simultaneously, the SKB method has the best performance with 52.2%, followed by the NER method with 51.2% and the VSM method with 43.6%.

We also investigated the sizes of top clusters corresponding to major people at the top $F_{measure\_top\_aver}$. The averaged sizes of the two top clusters in the 216 test sets for the SKB method are 44.1 and 23.3, while those for the NER method are 28.4 and 27.8, and those for the VSM method are 44.5 and 17.4. These cluster size numbers are close to the numbers of documents of major people that we have prepared.

5. Conclusion

In this research we focused on the problem of disambiguate personal name in web search results. To solve this problem, we have proposed a new method to measure the similarities between documents: similarity via knowledge base (**SKB**). Our method uses a knowledge base to find out topic words, which are important keywords in documents, in order to find out shared contexts of documents and to more easily calculate the weight of the shared contexts. Then, we use these similarity results for the agglomerative clustering to group related documents together. Our SKB method performed better than two traditional methods: the vector space model (VSM) method and the named entity recogniton (NER) method.

[References]

[1] T. Pedersen, A. Kulkarni, R. Angheluta, Z. Kozareva, T. Solorio. Name Discrimination by Clustering Similar Contexts. CICLing2005.
[2] R. Bekkerman, A. McCallum. Disambiguating Web Apprearances of People in a Social Network. WWW2005.
[3] R. Guha, A. Garg. Disambiguating People in Search. WWW2004.
[4] http://www.alias-i.com/lingpipe/
[5] http://www.dmoz.org/
[6] http://www.google.com/
[7] R. Baeza-Yates, B. Ribeiro-Neto. Modern Information Retrieval. Addison Wesley Longman Publishing 1999.
[8] C. D. Manning, H. Schutze. Foundations of Statistical Natural Language Processing. The MIT Press 2003.
[9] A. Bagga, B. Baldwin. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. ACL 1998.
[10] B. Malin. Unsupervised Name Disambiguation via Social Network Similarity. SIAM ICDM 2005.
[11] G. S. Mann, D. Yarowsky. Unsupervised Personal Name Disambiguation. Computational Natural Language Learning 2003.
[12] X. Wan, J. Gao, M. Li, B. Ding. Person Resolution in Person Search Results: WebHawk. CIKM 05.
[13] http://trec.nist.gov/
[14] http://dblp.uni-trier.de
[15] http://www.amazon.com
[16] http://www.tartarus.org/martin/PorterStemmer/
[17] H. Schutze. Automatic Word Sense Discrimination. Computational Linguistics, 24(1):97-123, 1998

Quang Minh Vu
Quang Minh Vu is a doctor student at the University of Tokyo. He received his B.E. from Kyoto University and M.E. from the University of Tokyo in 2003 and 2005, respectively. His research interests include information retrieval, text mining, and natural language processing.

Tomonari Masada
Dr. Masada is a postdoctoral researcher at the National Institute of Informatics. He received his PhD degree from the University of Tokyo in 2004. His research interests include text mining and information retrieval.

Atsuhiro Takasu
Atsuhiro Takasu received B.E., M.E. and Dr. Eng. from the University of Tokyo in 1984,1986 and 1989, respectively. He is a professor of National Institute of Informatics, Japan. His research interests are database systems and machine learning. He is a member of ACM, IEEE, IEICE, IPSJ and JSAI.

Jun Adachi
Jun Adachi is a professor of Digital Content and Media Sciences Research Division, National Institute of Informatics (NII), Japan. He is also the Director of Development and Operations Department of NII. His professional career was largely spent in research and development of NACSIS information systems, such as NACSIS-CAT and NACSIS-ELS. He is also an adjunct professor of the Graduate School of Information Science and Technology, University of Tokyo. His research interests are information retrieval, text mining, digital library systems, and distributed information systems. Adachi received a BE, ME and Doctor of Engineering in Electrical Engineering from University of Tokyo in 1976, 1978 and 1981, respectively. He is a member of IPSJ, IEICE, IEEE, and ACM.