Volume 44, issue 4, July 2008

ISSN 0306-4573

INFORMATION PROCESSING & MANAGEMENT

An International Journal

Email: ipm@unisi.ch
Web site: http://www.elsevier.com/locate/infoproman

Available online at www.sciencedirect.com

ScienceDirect

Information Processing and Management 44 (2008) 1546–1561

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Improving the performance of personal name disambiguation using web directories

Quang Minh Vu [a,b,*], Atsuhiro Takasu [b], Jun Adachi [b]

[a] *The University of Tokyo, Graduate School of Information Science and Technology, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan*
[b] *National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan*

Received 19 July 2007; received in revised form 28 October 2007; accepted 4 November 2007
Available online 26 December 2007

Abstract

Frequent requests from users to search engines on the World Wide Web are to search for information about people using personal names. Current search engines only return sets of documents containing the name queried, but, as several people usually share a personal name, the resulting sets often contain documents relevant to several people. It is necessary to disambiguate people in these result sets in order to to help users find the person of interest more readily. In the task of name disambiguation, effective measurement of similarities in the documents is a crucial step towards the final disambiguation. We propose a new method that uses web directories as a knowledge base to find common contexts in documents and uses the common contexts measure to determine document similarities. Experiments, conducted on documents mentioning real people on the web, together with several famous web directory structures, suggest that there are significant advantages in using web directories to disambiguate people compared with other conventional methods.

Keywords: Web search; Name disambiguation; Document similarity; Knowledge base; Web directories

1. Introduction

Searching for information about a person on the World Wide Web (WWW) is an increasing requirement in information retrieval. As the population of the WWW is increasing rapidly, the WWW has become the largest document resource ever seen. Search engines are effective tools to help users retrieve documents from such a huge database. Of the queries by users to search engines, a certain portion of queries, from 5% to 10%, includes people's names (Guha & Garg, 2004).

Search results returned from search engines for a personal name query often contain documents relevant to several people because a name is usually shared by several people. For example, in the top 100 results returned

[*] Corresponding author. Address: National Institute of Informatics, Adachi Laboratory, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan. Tel.: +81 80 5689 1359.
E-mail addresses: vuminh@nii.ac.jp (Q.M. Vu), takasu@nii.ac.jp (A. Takasu), adachi@nii.ac.jp (J. Adachi).

0306-4573/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2007.11.001

by the Google search engine[1] for the name query "Jim Clark", there are at least eight different Jim Clarks. Due to this name ambiguity problem, users have to manually investigate the result documents to filter out people in whom they have no interest.

In our research, we endeavor to disambiguate people cited in the result set by reranking documents according to their relevancies to a certain person. First, users notify the search engine their person of interest by selecting one document. Then, upon receiving the user's selection, our system will rerank documents in the result set by the relevance order to the selected document.

In previous studies, researchers focused on disambiguation of people in some types of documents, such as scientific publications (Han, Giles, Zha, Li, & Tsioutsiouliklis, 2004) or newspaper articles (Bagga & Baldwin, 1998). However, documents on the web have distinct characteristics that differ from scientific documents or news articles. Documents on the web are 'noisy', cover a broad range of topics, and come in various formats. In addition, the amount of information referring to a person varies from few sentences to the whole document. Therefore, previous approaches are limited when disambiguating people in web documents.

To disambiguate people in web documents, we propose a new method that uses web directories to improve the disambiguating performance. On the WWW, many web directories are created to be available freely for everyone, such as the Dmoz directory,[2] the Google directory,[3] and the Yahoo directory.[4] These are collections of web documents categorized into several sets on different topics. We use these sets of documents to enrich the extractable information in web documents. This enrichment enables us to determine the documents' topics and to extract the common contexts in document pairs more readily. The measure of the common contexts in document pairs is used to rerank the result documents.

The remainder of this paper is organized as follows. In Section 2, we summarize previous studies on name disambiguation. In Section 3, we present the details of our method. We indicate the limitations of the vector space model method and propose a new method to reduce these limitations. Then, in Section 4, we describe the name disambiguation system of our proposed method. Experimental results and comparisons with other methods are given in Section 5. We describe the advantages and limitations of our methods in Section 6. Finally, we present our conclusions in Section 7.

2. Related work

Bagga and Baldwin (1998) solved the problem of personal name disambiguation in news articles. They used an internal document co-reference system (Baldwin et al., 1995) to extract text relevant to a person in a document. Then, they used the vector space model (VSM) (Baeza-Yates & Ribeiro-Neto, 1999) to measure similarities between articles. In news articles, documents referring to the same person tend to describe a single event. Therefore, they usually have the same topic and the traditional VSM method can measure document similarities very well. However, people on the web may have several appearances related to different events. For example, a computer scientist may have different research interests overtime, so his or her publications may be about different research topics. Therefore, the specific topics of his or her publications may vary, even though they concern the same general topic of computer science. In such a case, where the topic relationship between documents is not strong, the VSM method may not measure document similarities adequately because there are few co-occurring terms among documents.

Pedersen, Kulkarni, Angheluta, Kozareva, and Solorio (2005) extracted contexts in documents to disambiguate people. They calculated the context of documents using a method called *second order context vectors* (Schutze, 1998). They applied the log likelihood method (Manning & Schutze, 2003) together with singular value decomposition (Manning & Schutze, 2003) for co-occurrence information to calculate context vectors of terms. Then, they defined the document context vector to be the average vector of context vectors of all terms in the document. Document context vectors were used to cluster documents into groups. In their research, they experimented with famous people, such as the soccer players Ronaldo and David Beckham,

[1] http://www.google.com.
[2] http://www.dmoz.org/.
[3] http://directory.google.com.
[4] http://dir.yahoo.com.

and the former Prime Minister of Israel, Shimon Peres. However, this approach may not work well when dealing with people who are not famous because only a few documents relate to them, which makes the building of context vectors difficult. Bollegala, Matsuo, and Ishizuka (2006) used the *C-value/NC-value* method (Frantzi, Ananiadou, & Tsujii, 1998) to extract key phrases related to people. Then, they sent key phrases as queries to search engines and built key phrases' contexts using snippets of the resulting documents. However, this method is expensive because it requires many query transactions to build contexts for key phrases.

To deal with ordinary people, who are included in few relevant documents, Bekkerman and McCallum (2005) proposed a method to extract a group of people simultaneously. People in this group are related to one another so their relevant web page set has a greater number of pages; these pages may share the same topic and be connected. The authors proposed two methods of extracting a group of people: one uses link information in web pages and the other uses the Agglomerative Conglomerative Double Clustering (A/CDC) algorithm (Slonim & Tishby, 2000) to group together web pages having the same topic. The use of this method is limited because, when we search for a person on the web, we may not know his or her social network in advance.

Several research groups have proposed several methods of extracting personal profiles, or entities related to people. Mann and Yarowsky (2003) used the pattern-matching method (Ravichandran & Hovy, 2001) to extract personal profiles (birthday, birthplace, occupation, etc.). Guha and Garg (2004) used databases, such as DBLP[5] and Amazon, [6] to extract authors' names and research keywords. These methods have some disadvantages as follows. The method of extracting personal profiles may not work well with web pages other than profile pages, while the method that uses a dictionary-like database cannot extract terms not listed in the database. Wan, Gao, Li, and Ding (2005) used natural language-processing techniques to extract named entities in documents. However, because web documents contain much noisy information, the extraction of named entities may not work well.

Previous researchers have targeted several types of documents: articles in newspapers, web documents of famous people, and web documents containing biographic data. However, as the number of documents disseminated on the WWW is growing dramatically, there are other types of web documents that have not yet been targeted. In addition to the growing number of web documents, there are numerous variations of document format and writing style. Therefore, previous approaches are limited when working with such documents. These limitations motivated us to develop a new method that can treat different kinds of web documents existing on the web. The characteristic mark of our approach is that we use a form of complement information to facilitate the procedure of feature extraction and document similarity measurement. Web directories have been used in our approach. They cover a greater range of topics than other types of resources used in previous research, such as the DBLP or the Amazon online bookshop, so they can work with documents that have different topics. In addition, several web directories containing a large number of documents already exist on the WWW, so the preparation cost is inexpensive.

3. Document similarities via a knowledge base

3.1. A review of the tf-idf weighting scheme

The vector space model (Baeza-Yates & Ribeiro-Neto, 1999) is the conventional method for measuring the similarity of two documents. In the vector space model, a document is represented by a feature vector formed from the weights of terms in the document. Here, we review the *tf-idf* term weighting scheme (Baeza-Yates & Ribeiro-Neto, 1999), which is a conventional approach often used in the vector space model. In the *tf-idf* term weighting scheme, term weights are calculated using the terms' occurrences in the document concerned and in a set of documents. If a term appears frequently in a document, then that term may be strongly related to the document concerned, so its weight should be proportional to its number of occurrences in the document. The

[5] http://dblp.uni-trier.de.

[6] http://www.amazon.com.

tf-idf weighting scheme also uses a term's occurrences in the document set to calculate term particularity. Intuitively, if a term appears frequently in many documents its particularity decreases.

Denote a set of N documents as $S_{doc} = \{doc_1, doc_2, \ldots, doc_N\}$, $df(t, S_{doc})$ is the number of documents in S_{doc} containing t. According to Zipf's law (Manning & Schutze, 2003), term particularity is proportional to $\log(\frac{N}{df(t,S_{doc})})$. Another derivation of term particularity using the information theory also arrives at the formula $\log(\frac{N}{df(t,S_{doc})})$ for term particularity (Aizawa, 2000). A term's weight is then calculated as follows:

$$idf(t, S_{doc}) = \log\left(\frac{N}{df(t, S_{doc})}\right) \tag{1}$$

$$tf\text{-}idf(t, doc, S_{doc}) = tf(t, doc) \times idf(t, S_{doc}) \tag{2}$$

where $tf(t, doc)$ is the number of times term t appears in the document doc.

The vector space model based on the *tf-idf* weighting scheme measures the similarity of two documents by using the inner product of document feature vectors. It works well when the two documents concern the same topic. When two documents concern the same topic, they have many common terms, so the inner product is large.

Although the *tf-idf* weighting scheme works well with documents on the same topic, it may not work well with documents relevant to the same person, as they have very few terms in common. There are two reasons for this. First, documents relating to the same person need not to be about the same topic. Rather, they may have slightly different specific topics under the same general topic; therefore, common terms between documents are rare. Second, because documents on the web contain noisy information, only text surrounding a person's name seems to be relevant to that person, not the whole document. This further reduces the number of common terms.

3.2. Measurement of term weights using a knowledge base

The *tf-idf* weighting scheme is limited when measuring documents relevant to the same person. We propose a new method that uses web directories to measure features of terms in a document (Vu, Masada, Takasu, & Adachi, 2007a). First, we give a brief introduction to web directories. Then, we propose two approaches using web directories to improve the measurement of term weights.

3.2.1. A knowledge base

As described in Section 3.1, text relevant to a person in a web document is short. Therefore, even keywords that are strongly related to that person have low term frequencies. To overcome this problem, we prepare several sets of documents, each set containing documents on the same topic and we use these sets of documents to measure term weights. We call such a collection a knowledge base, because it collects knowledge of several topics in several sets of documents. In our research, we use web directories in the role of a knowledge base. Below, we use "knowledge base" and "web directories" interchangeably to refer to a collection of documents on several topics and we use "a directory" to refer to a set of documents on the same topic. We name our method "Similarity via Knowledge Base (SKB)" to separate it from the vector space model based on the *tf-idf* term weighting scheme. Hereafter, we refer to the vector space model based on the *tf-idf* term weighting scheme as the traditional vector space model, or the VSM for its abbreviation.

3.2.2. Modification of term weight in documents

In a web document, text relevant to a person tends to be short because only a part of the document mentions the person and the web document may contain noise. Therefore, term weights calculated by Eq. (2) for keyword terms and for other terms differ only slightly.

Assume we have a directory whose topic is close to the document's topic. As the directory has abundant text, keywords related to the topic appear more frequently, so their term weights will be larger than the weights of other terms. It is reasonable to assume that keywords will appear on the web document as frequently as they appear in the directory if the relevant text on the web document increases in length. Therefore, we can use the large weights of keyword terms in the directory to amplify the small weights of keyword terms in the document.

Denote $S_{dir} = \{dir_1, dir_2, \ldots, dir_K\}$ as a set of web directories on several topics, M as the total number of documents in S_{dir}, $tf(t, dir_i)$ as the number of times term t appears in the directory dir_i, $df(t, S_{dir})$ as the number of documents in S_{dir} containing t, and length (dir_i) as the total number of word counts in the directory dir_i. We calculate term weights for the feature vector of directory dir_i as follows:

$$idf_{DIR1}(t, S_{dir}) = \log\left(\frac{M}{df(t, S_{dir})}\right) \tag{3}$$

$$tf\text{-}idf_{DIR1}(t, dir_i, S_{dir}) = \frac{tf(t, dir_i) \times idf_{DIR1}(t, S_{dir})}{\text{length }(dir_i)} \tag{4}$$

Because we have to compare feature vectors between directories in the next calculation step, we normalize term weights by dividing them by the lengths of the directories to facilitate comparison.

We modify the term weights in documents by taking the mean of the term weights calculated by Eqs. (2) and (4). We have tested the arithmetic mean and the geometric mean, and the geometric mean gives the better result of the two, because, when taking the arithmetic mean, terms that do not appear in the document have weights larger than zero and the total weight of these terms dominates the total weight of terms that appear in the document. Following on from this experimental result, we use the geometric mean in our research.

The details of term weight modification can be formalized as follows:

$$\begin{aligned} tf\text{-}idf_{\mathrm{SKB1}}(t, doc, dir_i) &= \sqrt{tf\text{-}idf(t, doc, S_{doc}) \times tf\text{-}idf_{DIR1}(t, dir_i, S_{dir})} \\ &= \sqrt{tf(t, doc)idf(t, S_{doc}) \times \frac{tf(t, dir_i)idf_{DIR1}(t, S_{dir})}{\text{length }(dir_i)}} \end{aligned} \tag{5}$$

Our modification of term weights functions analogously to a signal frequency filter. A document can be regarded as an information source and the set of all terms can be regarded as a range of frequencies. A document feature vector corresponds to a power spectrum, where a term weight corresponds to the power at a certain frequency. A directory will amplify weights of terms close to the topic in the directory while dampening the weights of the other terms.

3.2.3. Modification of term weight in directories

The *idf* factor in the *tf-idf* weighting scheme can be explained using the information entropy theory. For example, in Aizawa (2000), the author explained the $idf = \log\left(\frac{N}{df(t)}\right)$ factor for a term t as the information amount gained by t. Without the observation that t appears in a document d, d can be any document from a collection of N documents. However, given the fact that d contains t and there are *df* documents in the collection containing t, d is now chosen from *df* documents. Therefore, the information gained by t is the difference between the two entropies: $\log(\frac{1}{df}) - \log(\frac{1}{N}) = \log(\frac{N}{df})$.

We modify term weight measurements in the directories as follows (Vu, Masada, Takasu, & Adachi, 2007b). The explanation by Aizawa (2000) assumed that contexts of documents in the collection were independant to each other. Therefore, term t was assumed to be related with $df(t)$ different contexts in $df(t)$ documents. However, for our directories, documents in the same directory are supposed not to be independant to each other; some documents may have common contexts. If a term t that appears frequently in a certain directory but appears less frequently in general, then it tends to be strongly related to the directory's topic. Although t may have a large value of *df*, the number of contexts related to t should be much lower than *df* since documents containing t seem to have common context. Therefore, its gain of information amount should be increased.

We define the normalized document frequency of a term in a directory and in all directories as follows:

$$df(t, dir_i) = \frac{df(t, dir_i)}{M_i} \tag{6}$$

$$df(t, S_{dir}) = \frac{df(t, S_{dir})}{M} \tag{7}$$

where $df(t, dir_i)$ is the number of documents in dir_i containing term t, and M_i is the number of documents in dir_i.

We assume that when terms have normalized frequencies in a certain directory that are much larger than their normalized frequencies in all directories, then those terms appear to be topic terms in the directory concerned. We propose the following equations that can appropriately increase *idf* weights for topic terms:

$$modifier(t, dir_i) = \begin{cases} \frac{df(t,dir_i)}{df(t,S_{dir})}, & \text{if } \frac{df(t,dir_i)}{df(t,S_{dir})} > r \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

$$idf_{DIR2}(t, S_{dir}, dir_i) = \log\left(\frac{M}{df(t, S_{dir})} \times \text{modifier } (t, dir_i)\right) \tag{9}$$

where r is a given threshold that we call the document frequency ratio threshold.

We combine the modification Eq. (9) of term weights in directories with the modification Eq. (5) of term weights in documents and obtain the following equations to measure term weights:

$$tf\text{-}idf_{DIR2}(t, dir_i, S_{dir}) = \frac{tf(t, dir_i) \times idf_{DIR2}(t, S_{dir}, dir_i)}{\text{length } (dir_i)} \tag{10}$$

$$\begin{aligned} tf\text{-}idf_{\text{SKB2}}(t, doc, dir_i) &= \sqrt{tf\text{-}idf(t, doc, S_{doc}) \times tf\text{-}idf_{DIR2}(t, dir_i, S_{dir})} \\ &= \sqrt{tf(t, doc)idf(t, S_{doc}) \times \frac{tf(t, dir_i)idf_{DIR2}(t, S_{dir})}{\text{length } (dir_i)}} \end{aligned} \tag{11}$$

Our idea of using information from directory structure to modify term weights of topic terms has common points with the term weighting scheme using the term entropy with regard to an external directory structure in Kohonen et al. (2000). Both our approach and the approach in Kohonen et al. (2000) utilize the term probabilities in specific directories and in general directories to appreciate weights of topic terms. In Kohonen et al. (2000), training documents and test documents are from the same source and term weights for test documents are calculated using their entropies in training documents. However, in our approach, web directories and name ambiguous documents are from different sources, so we only use term weights in web directories to modify the term weight measurements in name ambiguous documents as above.

3.3. Measurement of document similarities

The measurement of document similarities is performed in two steps. First, we find directories that have topics close to that of the document. Then, we measure the document similarities using these selected directories. The details are as follows:

3.3.1. Find directories close in topic with the document

Because we do not know the documents' topics in advance, we have to guess their topics. For each document, we choose k directories in the knowledge base whose similarities to the document are the top k largest values. The similarity between a document d and a directory Dir is measured as follows:

$$\text{SIM}(doc, dir) = \sum_{t \in doc \cap dir} tf\text{-}idf_{\text{SKB}}(t, doc, dir) \tag{12}$$

where $tf\text{-}idf_{\text{SKB}}(t, doc, dir)$ is replaced by $tf\text{-}idf_{\text{SKB1}}(t, doc, dir)$ or $tf\text{-}idf_{\text{SKB2}}(t, doc, dir)$.

We call these top k directories of document doc the document's representative directories and denote this set of directories as $R(doc)$.

3.3.2. Measure document similarities

Denote a pair of documents as (doc_1, doc_2). For each directory dir_i in the union set $R(doc_1) \cup R(doc_2)$, we calculate the similarity between documents doc_1 and doc_2 via directory dir_i

$$\text{SIM}(doc_1, doc_2, dir_i) \sum_{t \in doc_1 \cap doc_2} tf\text{-}idf_{\text{SKB}}(t, doc_1, dir_i) \times tf\text{-}idf_{\text{SKB}}(t, doc_2, dir_i) \tag{13}$$

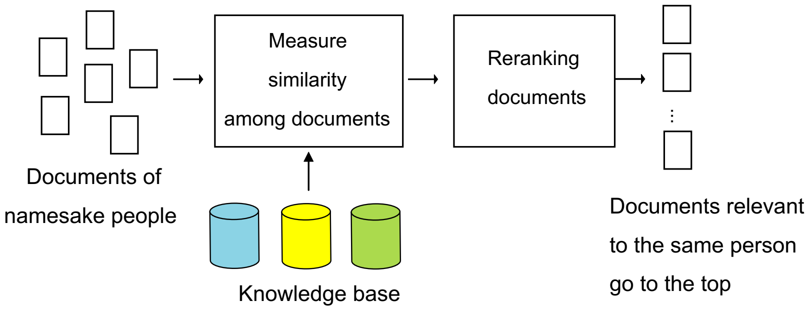

Fig. 1. Name disambiguation system.

After calculating the similarities of doc_1, doc_2 via all representative directories, we take their sum as the similarity of the document pair (doc_1, doc_2)

$$\mathrm{SIM}(doc_1, doc_2) = \sum_{dir_i \in R(doc_1) \cup R(doc_2)} \mathrm{SIM}(doc_1, doc_2, dir_i) \tag{14}$$

4. Name disambiguation system

Fig. 1 is an overview of our name disambiguation system. The system takes a knowledge base and documents of namesake people as input data. Then, it calculates document similarities between documents and helps users to find the desired person by reranking documents so that documents relevant to the person of interest go to the top of the list. The operational details are as follows:

(1) *Preprocessing documents*
We remove stop words and use the Porter stemming algorithm[7] to stem terms to their root forms. As web pages usually contain noisy information, only terms surrounding the personal name are considered as information strongly related to the people concerned. Therefore, after removing stop words, we select only n terms before and n terms after the personal names and create a collection of words containing information relevant to that person.

(2) *Calculation of document similarities*
We use a knowledge base to modify the documents' feature vectors using Eqs. (5) and (11). Then, we use Eqs. (13) and (14) to calculate document pair similarities. We denote the systems that use Eqs. (5) and (11) as SKB1 and SKB2, respectively.

(3) *Discrimination by reranking documents*
Our system uses a simple but effective reranking method, which was used by Guha and Garg (2004), to help users to discriminate between people in the result set. It is used as follows. First, users select from the result set a document that refers to the person of interest. Then, our system receives users' feedback information and reranks the result documents according to the order of their similarities to the selected document. Therefore, the reranking results show documents in the order of their relevance to the person the users are researching.

5. Experiments

5.1. Data sets

5.1.1. Documents of people

We selected 24 names as shown in the right column of Table 1. For each name, there was a particular person with that name who specialized in the research field shown in the left column of Table 1. We sent each

[7] http://www.tartarus.org/martin/PorterStemmer/.

Table 1
List of 24 name queries

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

name to the Google search engine and selected the top 100 results. In each result set, there were documents relevant to the person who specialized in the field shown in the left column of Table 1 and documents relevant to other people. The documents for all these people were used in our experiments. We removed documents that were not html documents. For each name, the person bearing that name and specializing in the field shown in the left column of Table 1 was associated with from 10 to 50 documents, whereas other people bearing that name were associated with between one and 10 documents. Table 2 shows the number of people and the number of relevant documents. The first and third columns show the number of relevant documents, the second and fourth columns show the number of people who had that number of relevant documents.

5.1.2. Creation of pseudo namesake document sets and real namesake document sets

In order to get a number of test data, we created name-ambiguous documents artificially as follows. We selected two result sets corresponding to the names of two people belonging to different research fields and mixed them together. Then, we replaced the personal names in the documents by the name X to create a set of documents of pseudo namesakes. In each mixed data set, there were two people with different professions, each with between 10 and 50 relevant documents. Besides these two people, there were several other people with between one and 10 relevant documents. For example, we mixed together the "Tom M. Mitchell" set containing several Mitchells and the "Paul G. Hewitt" set containing several Hewitts to create a document set that included documents referring to a computer scientist, a physicist, and several other people. As we had four research fields and six names in each research field, we could create $6 \times 6 \times \binom{4}{2} = 216$ combinations of names and produce 216 sets of pseudo namesake documents.

Besides experiments on pseudo namesake document sets, we also did carry experiments on real namesake document sets in order to verify the performance of our approach with real problems. The 24 real namesake document sets are result sets of 24 name queries shown in Table 1.

5.2. Web directory structures

We selected three well-known web directories on the WWW: the Google directory (http://directory.google.com), the Yahoo directory (http://dir.yahoo.com), and the Dmoz directory (http://dmoz.org). For each directory, we obtained all level two child nodes starting from the root node. Then, we selected direc-

Table 2
Numbers of documents of people

Number of relevant documents	Number of people	Number of relevant documents	Number of people
1	942	31–40	3
2–5	33	41–50	4
5–10	8	51–60	6
11–20	5	61–70	1
21–30	6		

Table 3
Number of directories and documents in directory structures

Directory name	Number of directories	Number of documents
Google10	214	6762
Google20	124	5318
Yahoo10	219	5979
Yahoo20	109	4524
Dmoz10	175	5701
Dmoz20	103	4551

tories with numbers of documents greater than a set threshold. We used threshold values of 10 and 20 to create six directory structures. The number of directories and the number of documents in the six directory structures are shown in Table 3.

5.3. Baseline methods

We compared our method with two conventional methods: VSM and named entity recognition (NER).

5.3.1. Vector space model method

In the VSM method, we removed stop words and stem words to their root form by using the Porter stemming algorithm. Then, we chose the terms inside the text windows centered at the personal name queries. We used Eq. (2) to calculate the weight of these terms and built the feature vectors of documents. We took the inner products of document feature vectors for the similarities between document pairs.

5.3.2. Named Entity Recognition method

In the NER method, we used the LingPipe software[8] to extract the entity names in the documents. Then, we used these names to construct feature vectors of the documents. The constituents of vectors were binary values (1 if a name appears in the document, 0 otherwise). We took the inner products of the document feature vectors for the similarities between documents.

5.4. Evaluation metrics

As described in Section 4, our system disambiguates people by reranking the result documents based on a document selected by the user. Therefore, we assumed that the user may choose any document doc_i in the result set, and evaluated the performance of the reranking result based on that document doc_i. We recorded the precision values at 11 recall points: 0%, 10%, 20%,...,90%, and 100% and denoted these as $P(doc_i, 0\%), P(doc_i, 10\%), P(doc_i, 20\%), \ldots, P(doc_i, 90\%), \text{and} P(doc_i, 100\%)$, respectively. We calculated the averaged precision values at these 11 recall points for all possible reranking sequences as follows:

$$P_{\text{aver-}doc}(k\%) = \frac{\sum_{doc_i} P(doc_i, k\%)}{\text{Number of documents in the result set}} \tag{15}$$

where $k = 0$, 10, 20,...,90, and 100.

We also took the averaged value of these 11 averaged precision values

$$P_{\text{aver}} = \frac{\sum_{k=0,10,\ldots,100} P_{\text{aver-}doc}(k\%)}{11} \tag{16}$$

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

5.5. Experimental results

In this section, we compare the experimental results of our SKB methods and those of baseline methods VSM and NER with the documents of people as described in the Section 5.1. Furthermore, we also investigate the robustness of our SKB methods over changes in directory structures and varying parameters. We applied six directory structures described in Section 5.1 to our SKB methods and investigated performance. We also varied the window size parameter n, and the number of representative directories parameter k to verify the robustness of SKB methods. We experimented with the document frequency ratio threshold in Eq. (8), $r = 1, 2, 5, 10$, the window size parameter, $n = 10, 20, 30, \ldots, 90$, and 100, and the number of representative directories parameter, $k = 10, 20$, and 30.

5.5.1. The overall performance for each method

Figs. 2–7 show the precision–recall graphs for the SKB methods using different directory structures and their comparisons with the baseline methods. Tables 4 and 5 show the comparison in terms of the averaged precision value P_{aver} between the baseline methods VSM, NER and our proposed methods SKB1 and SKB2. In this experiment, we set the window size $n = 50$ and the number of representative directories $k = 20$. We set the frequency document ratio threshold for SKB2 $r = 5$. As can be seen from these Tables, our SKB1 and SKB2 methods together with six different directory sets outperform the baseline methods VSM and NER.

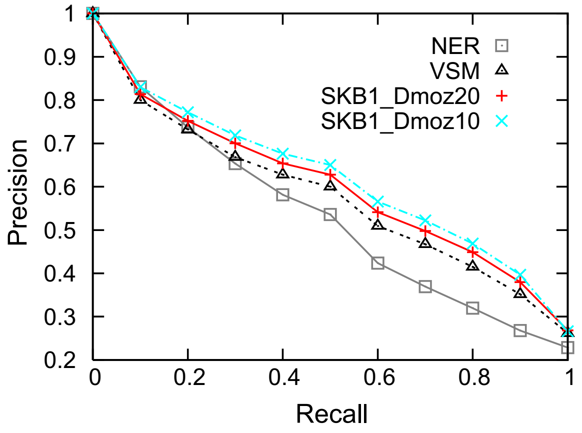

Fig. 2. Performance of SKB1 with Dmoz directories.

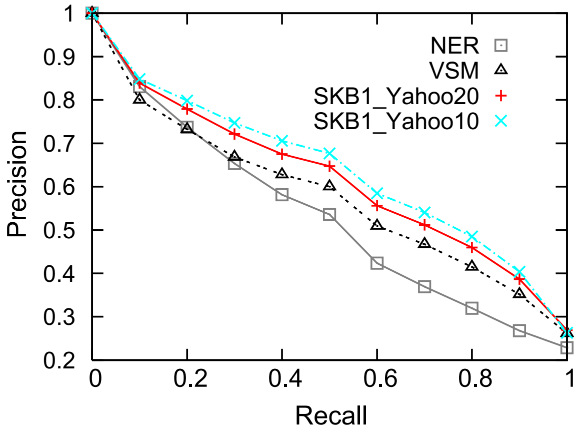

Fig. 3. Performance of SKB1 with Yahoo directories.

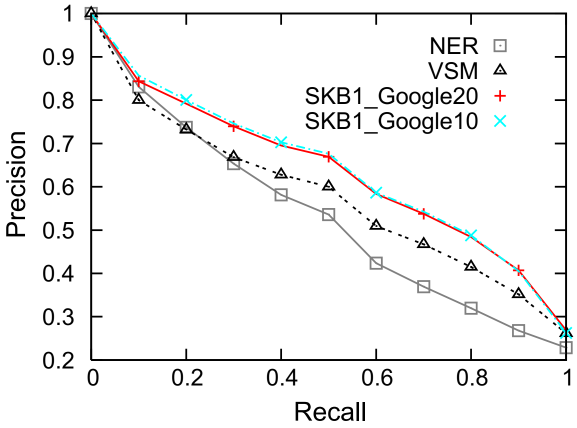

Fig. 4. Performance of SKB1 with Google directories.

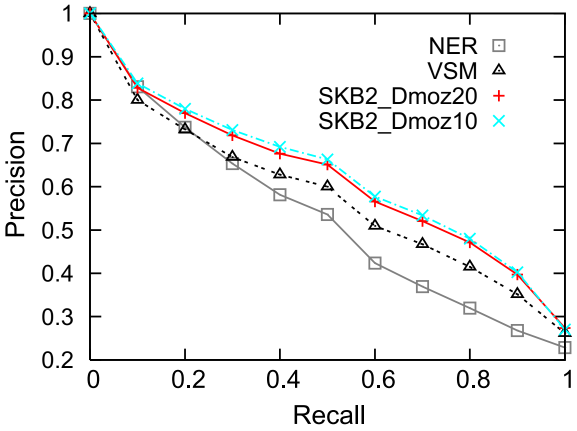

Fig. 5. Performance of SKB2 with Dmoz directories.

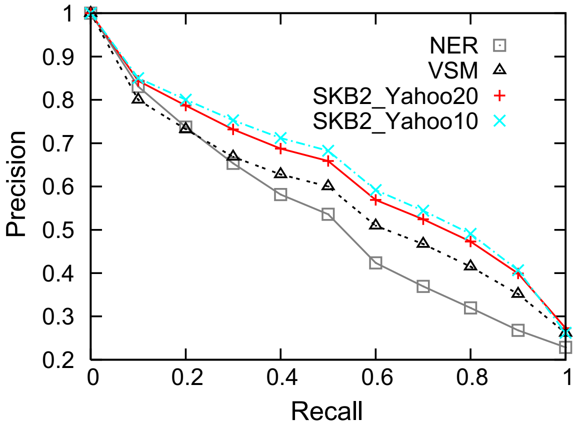

Fig. 6. Performance of SKB2 with Yahoo directories.

5.5.2. Performance of SKB2 when varying the document frequency ratio threshold

We experimented with SKB2 using different threshold values for document frequency ratio threshold: $r = 1, 2, 5, 10$. The directory structures used in these experiments were Dmoz10, Google10, and Yahoo10. Table 6 shows the experimental results. As can be seen from this table, SKB2 achieves good performances,

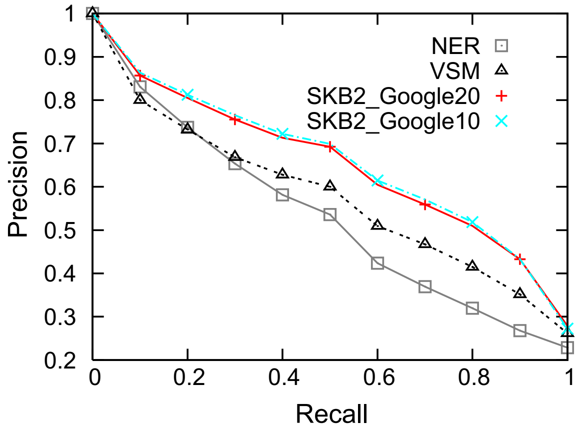

Fig. 7. Performance of SKB2 with Google directories.

Table 4
Comparison between VSM, NER and SKB1

Method	P_{aver} (%)
VSM	58.5
NER	54.1
SKB1_Google10	**64.2**
SKB1_Google20	63.8
SKB1_Yahoo10	64.1
SKB1_Yahoo20	62.2
SKB1_Dmoz10	62.4
SKB1_Dmoz20	60.8

Table 5
Comparison between VSM, NER and SKB2

Method	P_{aver} (%)
VSM	58.5
NER	54.1
SKB2_Google10	**66.1**
SKB2_Google20	65.5
SKB2_Yahoo10	64.5
SKB2_Yahoo20	63.2
SKB2_Dmoz10	63.4
SKB2_Dmoz20	62.5

Table 6
Average precisions of SKB2 with different threshold values of document frequency ratio

Directory	SKB1	SKB2, $r = 1$ (%)	SKB2, $r = 2$ (%)	SKB2, $r = 5$ (%)	SKB2, $r = 10$ (%)
Dmoz10	62.4	62.5	62.4	63.4	**63.6**
Google10	64.2	65.3	65.4	**66.1**	66.0
Yahoo10	64.1	64.7	**64.8**	64.5	64.4

especially when $r = 5$ and 10. This result agrees with the characteristic that the stronger the relation to a topic that a term has, the larger the document frequency ratio it has.

Table 7
Performance of SKB1 method with different window sizes

Window size	P_{aver} (%)
10	62.6
20	64.8
30	65.1
40	64.9
50	65.1
60	65.4
70	65.3
80	**65.5**
90	65.4
100	65.4

5.5.3. Performance of SKB systems when varying the window size

Tables 7 and 8 show the performance variations with different window size parameters. In these experiments, we used the Google20 directory structure with the number of representative directories set to 10. As can be seen from the results in these two tables, the SKB1 and SKB2 methods achieve better performance when the window size increases. We also experimented with the VSM method with different window size parameters. As shown in Table 9, we noted that the performance values of the VSM method decreased slightly when we increased the window size.

From the performance value decrease of the VSM method, we learn that the further the text is from the personal names, the more noise it contains. On the other hand, from the increased performance values of the SKB methods, we found that the SKB methods can effectively filter out noisy text and select relevant text far from the personal names.

Table 8
Performance of SKB2 method with different window sizes

Window size	P_{aver} (%)
10	63.4
20	66.1
30	66.3
40	66.4
50	66.4
60	66.65
70	66.69
80	**66.75**
90	66.73
100	66.68

Table 9
Performance of VSM method with different window sizes

Window size	P_{aver} (%)
10	**59.1**
20	58.6
30	58.6
40	58.6
50	58.5
60	58.4
70	58.3
80	58.4
90	58.4
100	58.3

Table 10
Performance of SKB1 with different number of representative directories

Number of representative directories	P_{aver} (%)
10	**65.1**
20	63.8
30	62.0

Table 11
Performance of SKB2 with different number of representative directories

Number of representative directories	P_{aver} (%)
10	**66.4**
20	65.5
30	63.5

5.5.4. Performance of SKBs when varying the number of representative directories

Tables 10 and 11 show the different performances with different number of representative directories $k = 10, 20, 30$. In this experiment, we used the Google20 directory structure with the window size fixed at 50. We recognize from the results shown in these two Tables that SKB1 and SKB2 methods achieved improved performance when the numbers of representative directories changed from 30 to 20 and 10.

5.5.5. Performance for each method on real namesake document sets

Table 12 shows the performance comparisons between VSM, NER, and SKB2 in the experiments on real namesake document sets. We use the averaged precision values at 11 recall points for all possible reranking sequences in the comparisons. The results show that our SKB2 performs best in 18 sets, follows by the

Table 12
Performance for each method on real namesake document sets

Name query	NER (%)	VSM (%)	SKB2 (%)
Adachi Jun	59.0	59.5	**64.2**
Sakai Shuichi	73.8	77.1	**79.7**
Tom M. Mitchell	71.9	79.9	**81.5**
Tanaka Katsumi	**79.5**	68.6	72.4
John D. Lafferty	76.9	81.4	**89.7**
Andrew McCallum	83.5	84.8	**88.5**
Paul G. Hewitt	64.2	69.5	**72.2**
Edwin F. Taylor	74.6	74.1	**85.6**
Paul W. Zitzewits	**85.7**	84.2	83.8
Frank Bridge	**55.8**	50.9	55.1
Kenneth W. Ford	52.6	51.9	**72.0**
Michael A. Dubson	**73.1**	70.1	72.5
Scott Hammer	76.2	68.6	**82.0**
Thomas F. Patterson	57.1	65.0	**81.5**
Michele L. Pearson	53.2	57.0	**64.4**
Henry F. Chambers	54.2	56.6	**64.2**
David C. Hooper	44.3	52.5	**60.0**
Lindsay E. Nicolle	83.1	85.7	**88.8**
John M. Roberts	76.9	74.0	**81.1**
David Reynolds	69.9	69.4	**72.8**
Thomas E. Woods	**91.4**	90.7	84.5
Thomas A. Brady	63.4	57.1	**63.8**
William L. Cleveland	60.4	59.0	**80.8**
Peter Haugen	50.2	**59.7**	59.4
Average	67.6	68.6	**75.1**

NER performs best in 5 sets, and the VSM performs best in 1 sets. On average, our SKB2 also outperforms the baseline methods VSM and NER.

6. Discussion

In this section, we describe how we exploited the web directories. We also note the advantages and the disadvantages of our method that uses web directories when disambiguating people.

Disambiguation of people in web documents is challenging because web documents are published by resources of different kinds, and useful information is mixed with noise. To improve effectiveness when processing web documents, we propose a new method that uses web directories to aid the extraction of the documents' features and the measurement of documents' similarities. We use information from directories to improve the calculation of the vector space model. The key to our approach is that web directories provide information about the relationship between directories' documents themselves and the relationship between directories' documents and other documents; these relationships cannot be found in the conventional vector space model method. We have proposed two approaches to exploit information from web directories. First, by investigating the relationship between documents referring ambiguous personal names and the documents on the web directories, we can improve the measurement of term frequencies in a document. Compared with the vector space model method and the named entity recognition method, we have improved the averaged precisions from 3.9% to 9.7%, and from 12.4% to 18.7%, respectively. Furthermore, we can exploit the relationship between the documents in the same web directories. Using this relationship, we can differentiate topic terms from common terms, even if they have the same characteristic in that they appear frequently in some documents. This exploitation can be regarded as an attempt to measure topic frequencies of terms. Although, we cannot count topic frequencies precisely, we can use web directories to modify term frequencies in documents to approach topic frequencies. This second exploitation results in a further improvement of averaged research precision from 6.8% to 12.9%, and from 15.5% to 22.2% over the VSM method and over the NER method, respectively.

We investigated the robustness of our approaches over changes of directory structure and variation of system parameters. The experimental results with different directory structures and different system parameters support the conclusion that our SKB methods achieve stable performance.

From the practical point of view, our approach has advantages as well as limitations. The greatest advantage is that it requires little preparation because the existing web directories can be used directly with virtually no preprocessing. In addition, the broad coverage of web directory topics enables the use of our approach with a broad range of people. On the other hand, the most significant limitation of our approach is its increasing cost of computation. The increase is proportional to the number of directories used.

7. Conclusions

Disambiguation of people in web searches is an increasing requirement for the new trends in web search systems. We propose a new method that uses web directories as a knowledge base to improve the disambiguation performance. Using web directories, we propose two approaches to better measure term weights. We have experimented with our approaches using several existing web directories to disambiguate documents of people on the web. The results showed a significant improvement with our system over the conventional methods: the vector space model method and the named entity recognition method. We also verified the robustness of our methods experimentally with different web directory structures and with different parameter values.

References

Aizawa, A. (2000). The feature quantity: An information theoretic perspective of tfidf-like measures. In *SIGIR'00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 104–111). ACM Press: New York, NY, USA.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley Longman Publishing.

Bagga, A., & Baldwin, B. (1998). Entity-based cross-document coreferencing using the vector space model. In ACL1998.

Baldwin, B., Collins, M., Eisner, J., Ratnaparkhi, A., Rosenzweig, J., & Sarkar, A. (1995). University of pennsylvania: Description of the University of Pennsylvania system used for muc-6. In *MUC6'95: Proceedings of the 6th conference on Message understanding* (pp. 177–191). Association for Computational Linguistics: Morristown, NJ, USA.

Bekkerman, R., & McCallum, A. (2005). Disambiguating web appearances of people in a social network. In *The fourteenth international World Wide Web conference, WWW2005*.

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2006). Extracting key phrases to disambiguate personal name queries in web search. In *Proceedings of the workshop "How can Computational Linguistics improve Information Retreival?"*, COLING-ACL 2006.

Frantzi, K.T., Ananiadou, S., & Tsujii, J. (1998). The c-value/nc-value method of automatic recognition for multi-word terms. In *ECDL'98: Proceedings of the second European conference on research and advanced technology for digital libraries* (pp. 585–604). Springer-Verlag: London, UK.

Guha, R., & Garg, A. (2004). Disambiguating people in search. In *The thirteenth international World Wide Web conference, WWW2004*.

Han, H., Giles, L., Zha, H., Li, C., & Tsioutsiouliklis, K. (2004). Two supervised learning approaches for name disambiguation in author citations. In *JCDL'04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries* (pp. 296–305). ACM Press, New York, NY, USA.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Paatero, V., Saarela, A., et al. (2000). Organization of a massive document collection. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, 11*(3), 574–585.

Mann, G. S., & Yarowsky, D. (2003). Unsupervised personal name disambiguation. In *Proceedings of computational natural language learning 2003*.

Manning, C. D., & Schutze, H. (2003). *Foundations of statistical natural language processing*. The MIT Press.

Pedersen, T., Kulkarni, A., Angheluta, R., Kozareva, Z., & Solorio, T. (2005). Name discrimination by clustering similar contexts. In *Proceedings of the sixth international conference on intelligent text processing and computational linguistics*.

Ravichandran, D., & Hovy, E. (2001). Learning surface text patterns for a question answering system. In *ACL'02: Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 41–47). Association for Computational Linguistics, Morristown, NJ, USA.

Schutze, H. (1998). Automatic word sense discrimination. *Computational Linguistics, 24*(1), 97–123.

Slonim, N., & Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. In *SIGIR'00: Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 208–215). ACM Press: New York, NY, USA.

Vu, Q.M., Masada, T., Takasu, A., & Adachi, J. (2007a). Using a knowledge base to disambiguate personal name in web search results. In *SAC'07: Proceedings of the 2007 ACM symposium on Applied computing* (pp. 839–843). ACM Press: New York, NY, USA.

Vu, Q.M., Masada, T., Takasu, A., & Adachi, J. (2007b). Using web directories for similarity measurement in personal name disambiguation. In *AINA workshop/symposia, the 2007 IEEE international symposium on data mining and information retrieval* (Vol. 1, pp. 379–384). IEEE Computer Society: Los Alamitos, CA, USA.

Wan, X., Gao, J., Li, M., & Ding, B. (2005). Person resolution in person search results: Webhawk. In *Proceedings of the ACM fourteenth conference on information and knowledge management, CIKM2005*.

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.

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 is a professor of Digital Content and Media Sciences Research Division, National Institute of Informatics (NII), Japan. He is also an adjunct professor of the Graduate School of Information Science and Technology, University of Tokyo. He is a member of IPSJ, IEICE, IEEE, and ACM.