Special issue: Leading ICT technologies in the Information Explosion

Research Paper

Utilization of external knowledge for personal name disambiguation

Quang Minh VU[1], Atsuhiro TAKASU[2] and Jun ADACHI[3]
[1,2,3]*National Institute of Informatics*

ABSTRACT

The amount of information on the World Wide Web (WWW) is increasing at an explosive rate, and the role of computer systems in processing such a huge amount of data has become crucial. In this paper, we focus on the name disambiguation problem when searching for people, because information about people is an important part of the web and improvements to personal information may benefit many web citizens. The name ambiguity problem occurs frequently when searching for people, because a name may be shared by several people. In this research, we use external knowledge while solving this problem, so that we can analyze information in web documents more easily. We collect web directories and use the latent Dirichlet allocation method to extract latent topics from web directories. The extracted topics are used to modify the search result documents so that important contexts that help to discriminate people can be recognized more easily. We carried out experiments with real web documents and verified the advantages of our approach over other disambiguation approaches that use the vector space model and named entity recognition methods.

KEYWORDS

Personal name disambiguation, knowledge base, latent Dirichlet allocation, latent topic extraction, document similarity

1 Introduction

Text documents on the World Wide Web (WWW) are rapidly increasing in number. It is said that the information amount on the web is doubling every few years. Users may be overwhelmed by such a huge amount of documents, and this trend raises a question of how we can exploit the WWW effectively.

We focused on information on the WWW that relates to people and developed a new method that attempts to process web documents and filter useful information to users. Searching for information related to a certain person is an everyday need for most web users. They usually use personal names when searching for people. However, in many cases, querying by personal name does not give satisfactory results. Since many people may have the same name, the results may contain information that is relevant to other people besides the person of interest. In order to filter useful information, we can use a re-ranking method whereby the search system interacts with users to get feedback to re-rank documents. First, users select a correct document for search system to use to re-rank its results so that correct documents will move to the top and users can find them more easily.

Some difficulties in processing web documents in searching are the sparseness and noisiness of information related to people. There are often very few documents related to individuals, so finding an appropriate one would be like finding a needle in a haystack. Moreover, in many web documents, only a small fraction of the information is directly related to the person of interest. That is, useful information is mixed with useless information in the same document.

The key point to our research is that we use external knowledge, i.e., knowledge that is external to the search's retrieved documents, to re-rank the results. We use web directories as the external knowledge in order

Received September 1, 2008; Revised December 1, 2008; Accepted December 2, 2008.
[1] *vuminh@nii.ac.jp*, [2] *takasu@nii.ac.jp*, [3] *adachi@nii.ac.jp*
DOI: 10.2201/NiiPi.2009.6.3

to complement sparse information and to separate useful information from useless information. Web directories contain information on many topics so that we can select a suitable topic from them that is close to the topic in a web document. Our basic idea is to use a set of topics from web directories to recognize important topics in web documents. A benefit of using topics from web directories is that the directories have a large amount of text so that their topics appear more often than those in normal web documents. Therefore, by using these strong topics, our method can filter out noisy information more easily and can recognize topics in normal web documents more effectively.

The outline of our approach is as follows. First, we use the latent Dirichlet allocation method [5] [8] to extract topics in web directories. Next, we use the extracted topics to model the topics of new web documents and modify the new documents so that important terms that are related to documents' main topics have more weight. After that, we calculate the common parts of the modified documents to measure their similarities. These similarity values are used in the re-ranking step to bring useful documents toward the top of the results list.

The rest of this paper is organized as follows. Section 2 summarizes related research on personal name disambiguation. Section 3 introduces our approach. Section 4 and 5 describe experiments we conducted and their results. Section 6 discusses the advantages and disadvantages of our approach in light of these results. We give our conclusions in Section 7.

2 Related researches

The basic task in name disambiguation is to recognize important information related to people in documents. In previous researches, approaches to recognizing such information can be roughly classified into two groups. In one group, researchers only use documents having ambiguous names to extract the important information. In the other group, researchers use external information in addition to documents having ambiguous names to improve disambiguation results. In this section, we review these two approaches.

2.1 Utilization of internal information

Vector space model approach

To solve the problem of disambiguating personal names in newspapers [3], researchers have used the vector space model (VSM) [2] approach to build bags of words for documents and to disambiguate personal names. Newspaper articles are filled with much information related to people, and the bag of words method is able to construct contexts of people.

Personal context building approach

In order to build contexts of people more effectively, researchers have applied the second-order context vector method [13] to web documents to disambiguate names [11]. In this approach, co-occurrences of terms are used to build a co-occurrence matrix, and the context of a term is represented by a set of terms that co-occur frequently with that term. The contexts of terms in a document are then summarized to create a context for the document. These document context vectors are called second-order context vectors, and they can represent the contexts of people and disambiguate people.

2.2 Utilization of external information

Some approaches have tried to use external information to complement the sparse information in order to improve disambiguation performance [4] [6] [9] [17].

Search engine snippet utilizing approach

In [6], researchers used the C/NC value method [7] to extract key phrases from documents. They put the extracted key phrases in search-engine queries and used snippets from search results to build the key phrases' contexts. These contexts were then used to represent the contexts of people for name disambiguation.

Social network utilizing approach

Social networks have been used for disambiguation. In [9], researchers built up personal relationships from a movie database. They used a graph analyzing algorithm to group name occurrences that had similar interactions with other names. In [4], the researchers assumed that someone's community was known in advance and tried to identify a person bearing an ambiguous name and his/her community at the same time. They identified the community of that person to get more relevant information, and consequently, disambiguation performance improved.

Named Entity Recognition approach

The named entity recognition (NER) approach [12] was used in several studies to extract personal information [10] [17]. NER requires training data to train a recognition algorithm, and this training data can also be regarded as a kind of external information.

2.3 Advantages and disadvantages of previous researches

The previous researches targeted documents in certain applications and proposed approaches that fit the specific characteristics of documents in each application. However, it is difficult to extend these approaches to more general web documents. The VSM and second-order context approaches work well with documents that have rich information but are limited when only sparse information is available. NER can extract en-

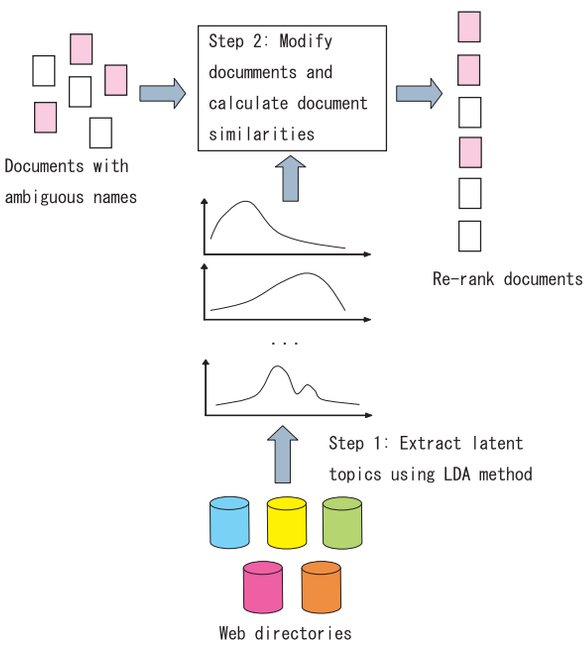

Fig. 1 Overview of our approach.

tity names from well-formatted documents, but it is not robust enough to deal with noisy documents. Other approaches that utilize social network information are difficult to extend to an arbitrary person because information about his/her social network might not be known.

3 Our approach

3.1 Web directories as external knowledge

Information sparseness and noise in the web are difficulties facing the task of name disambiguation from web pages. Our approach to tackle these difficulties is to use external information to enrich the web page content. We chose web directories as the external knowledge sources because they have certain advantages. First, web directories cover a large number of topics, so that we can find appropriate topics from them to enrich personal contexts in general web documents. Second, web directories are carefully prepared and organized; we can expect that they would be less noisy than general web documents.

preprocess web directories and used information from web directories to disambiguate personal names. The outline of what we did is shown in Fig. 1. First, we preprocess web directories to extract the latent topics from them. The topics of web directories are mixed together in documents, so we have to extract topics from the web directories and to calculate the distribution of words for each topic. Next, we use the extracted latent topics to recognize important topics in new web documents. We compare these latent topics with words in the new document to find topics that are close to that document and then modify that document so that its main topics becomes more evident. These modifications help us to filter out useful documents and to find common contexts among documents of the search results. We call our method Similarity via Knowledge Base with Latent Dirichlet Allocation (SKB-LDA). The following subsections describe the entire process in detail.

3.2 Extraction of topics from web directories

The LDA method extracts latent topics from a collection of documents. We modified the hyper-parameter settings of the original LDA so that it would be more adaptive to web directories. Below, we summarize the LDA method and describe our modifications to the hyper-parameter settings.

Overview of LDA

The LDA method analyzes the topics contained in a collection of documents. It assumes that a document's words are created in two steps. For each word slot, it first selects a topic for the word according to the topic distribution of the document. It then generates the word according to the word distribution of the selected topic. This two-step generation process is used independently to generate each word in document.

The mathematical notation for the generation process is as follows. Let T, D, and W be the number of topics, the number of documents, and the number of words, respectively. The topic distribution of a document d is parameterized by a vector $\Theta_d = (\vartheta_{d,1}, \vartheta_{d,2}, \cdots, \vartheta_{d,T})$. This topic distribution vector is assumed to be selected from a Dirichlet distribution as follows.

$$P(\Theta_d|\vec{\alpha}) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \vartheta_{d,1}^{\alpha_1-1} \vartheta_{d,2}^{\alpha_2-1} \cdots \vartheta_{d,T}^{\alpha_T-1} = \frac{1}{\Delta(\vec{\alpha})} \vartheta_{d,1}^{\alpha_1-1} \vartheta_{d,2}^{\alpha_2-1} \cdots \vartheta_{d,T}^{\alpha_T-1} \quad (1)$$

where $\vec{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_T)$ is a hyper-pameter vector of the Dirichlet distribution and $\Delta(\vec{\alpha}) = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)}$. The word distribution of a topic t is also parameterized by a vector $\Phi_t = (\varphi_{t,1}, \varphi_{t,2}, \cdots, \varphi_{t,W})$ that is also selected from a Dirichlet distribution as follows.

$$P(\Phi_t|\vec{\beta}) = \frac{\Gamma(\sum_{i=1}^{W} \beta_i)}{\prod_{i=1}^{W} \Gamma(\beta_i)} \varphi_{t,1}^{\beta_1-1} \varphi_{t,2}^{\beta_2-1} \cdots \varphi_{t,W}^{\beta_W-1} = \frac{1}{\Delta(\vec{\beta})} \varphi_{t,1}^{\beta_1-1} \varphi_{t,2}^{\beta_2-1} \cdots \varphi_{t,W}^{\beta_W-1} \quad (2)$$

where $\vec{\beta} = (\beta_1, \beta_2, ..., \beta_W)$ is a hyper-parameter vector of the Dirichlet distribution and $\Delta(\overrightarrow{\beta}) = \frac{\prod_{i=1}^{W}\Gamma(\beta_i)}{\Gamma(\sum_{i=1}^{W}\beta_i)}$.

Modifications to hyper-parameter setting

Web directories are not arbitrary collections of documents. In an arbitrary collection, documents are not well categorized, whereas each set of documents in our web directories contains documents that are close in topic to each other. In order to model this difference better, we modified the hyper-parameter settings as follows. Since each directory contains documents that are close in topic, we chose one topic distribution for all documents in the same directory. The number of topics was set to be equal to the number of directories and topic distribution in a directory was biased to its corresponding topic. To model this bias, we selected different hyper-parameter vectors for different directories. A hyper-parameter vector for directory $\mathcal{D}$ has a large $\alpha_{\mathcal{D},\mathcal{D}} = k\alpha$ for a topic $t_{\mathcal{D}}$, whereas it has a small $\alpha_{\mathcal{D},j} = \alpha$ for other topics t_j, $j \neq \mathcal{D}$.

$$\vec{\alpha}^{(i)} = (\alpha_{\mathcal{D},1} = \alpha, \cdots, \alpha_{\mathcal{D},\mathcal{D}} = k\alpha, \cdots, \alpha_{\mathcal{D},T} = \alpha) \tag{3}$$

Estimation of parameters

We used the Gibbs sampling method to estimate documents' topic distribution vectors Θ_i and topic word distribution vectors Φ_t. The Gibbs sampling algorithm tries to assign a topic ID to each word. First, it randomly assigns a topic ID to each word. Then, it repeatedly updates the topic ID for each word until convergence. To update the topic ID of a word w_i, the algorithm needs a formula to calculate its topic distribution from the topic IDs of other words. We derive this formula as follows.

Denote $\overrightarrow{w} = (w_1, w_2, ..., w_L)$ to be a vector composed by lining up all words in all documents. Denote t_i as the topic ID assigned to word w_i and $\overrightarrow{t} = (t_1, t_2, ..., t_L)$ as the vector of topic IDs. Also let $\overrightarrow{w^{-i}} = (w_1, w_2, ...w_{i-1}, w_{i+1}, ...w_L)$ and $\overrightarrow{t^{-i}} = (t_1, t_2, ...t_{i-1}, t_{i+1}, ...t_L)$. The probability of assigning t_i to word w_i is as follows.

$$\begin{aligned} P(t_i|\overrightarrow{w}, \overrightarrow{t^{(-i)}}) &= P(t_i|w_i, \overrightarrow{w^{(-i)}}, \overrightarrow{t^{(-i)}}) \\ &= \frac{P(t_i, w_i|\overrightarrow{w^{(-i)}}, \overrightarrow{t^{(-i)}})}{P(w_i|\overrightarrow{w^{(-i)}}, \overrightarrow{t^{(-i)}})} \\ &\propto P(t_i, w_i|\overrightarrow{w^{(-i)}}, \overrightarrow{t^{(-i)}}) \end{aligned} \tag{4}$$

$$P(t_i, w_i|\overrightarrow{w^{(-i)}}, \overrightarrow{t^{(-i)}}) = \frac{P(\overrightarrow{w}, \overrightarrow{t})}{P(\overrightarrow{w^{(-i)}}, \overrightarrow{t^{(-i)}})}$$

$$= \frac{P(\overrightarrow{w}|\overrightarrow{t})}{P(\overrightarrow{w^{(-i)}}|\overrightarrow{t^{(-i)}})} \cdot \frac{P(\overrightarrow{t})}{P(\overrightarrow{t^{(-i)}})} \tag{5}$$

In Eq. (4), $P(w_i|\overrightarrow{w^{(-i)}}, \overrightarrow{t^{(-i)}})$ is the same for all t_i.

For a directory $\mathcal{D}$ that contains w_i, we have:

$$\begin{aligned} P(\overrightarrow{t}) &= \int d\Theta P(\Theta_{\mathcal{D}}|\overrightarrow{\alpha_{\mathcal{D}}}) P(\overrightarrow{t}|\Theta_{\mathcal{D}}) \\ &= \frac{1}{\Delta(\overrightarrow{\alpha_{\mathcal{D}}})} \int d\Theta \prod_{t=1}^{T} \vartheta_{\mathcal{D},t}^{\alpha_{\mathcal{D},t}-1} \prod_{j=1}^{l} \vartheta_{\mathcal{D},t_l} \\ &= \frac{1}{\Delta(\overrightarrow{\alpha_{\mathcal{D}}})} \int d\Theta \prod_{t=1}^{T} \vartheta_{\mathcal{D},t}^{\alpha_{\mathcal{D},t}+n_{\mathcal{D},t}-1} \\ &= \frac{\Delta(\overrightarrow{\alpha_{\mathcal{D}}} + \overrightarrow{n_{\mathcal{D}}})}{\Delta(\overrightarrow{\alpha_{\mathcal{D}}})} \end{aligned} \tag{6}$$

$$P(\overrightarrow{t^{(-i)}}) = \frac{\Delta(\overrightarrow{\alpha_{\mathcal{D}}} + \overrightarrow{n_{\mathcal{D}}^{(-i)}})}{\Delta(\overrightarrow{\alpha_{\mathcal{D}}})} \tag{7}$$

where $n_{\mathcal{D},t}$ is the number of words in $\mathcal{D}$ to be assigned to topic t, $n_{\mathcal{D},t}^{(-i)}$ is the count number of words other than w_i in $\mathcal{D}$ to be assigned to topic t, $\overrightarrow{n_{\mathcal{D}}} = \{n_{\mathcal{D},t}\}_{t=1}^{T}$, and $\overrightarrow{n_{\mathcal{D}}^{-i}} = \{n_{\mathcal{D},t}^{-i}\}_{t=1}^{T}$.

Therefore,

$$\begin{aligned} \frac{P(\overrightarrow{t})}{P(\overrightarrow{t^{(-i)}})} &= \frac{\Delta(\overrightarrow{\alpha_{\mathcal{D}}} + \overrightarrow{n_{\mathcal{D}}})}{\Delta(\overrightarrow{\alpha_{\mathcal{D}}} + \overrightarrow{n_{\mathcal{D}}^{(-i)}})} \\ &= \frac{\alpha_{\mathcal{D},t_i} + n_{\mathcal{D},t_i}^{(-i)}}{\sum_t (\alpha_{\mathcal{D},t} + n_{\mathcal{D},t}^{(-i)})} \end{aligned} \tag{8}$$

We also have:

$$\begin{aligned} P(\overrightarrow{w}|\overrightarrow{t}) &= \int P(\overrightarrow{w_{\mathcal{D}}}|\overrightarrow{t}, \hat{\Phi}) P(\hat{\Phi}|\overrightarrow{\beta}) d\hat{\Phi} \\ &= \int \prod_{i=1}^{L} \varphi_{t_i,w_i} \prod_{t=1}^{T} \left(\frac{1}{\Delta(\overrightarrow{\beta})} \prod_{w=1}^{W} \varphi_{t,w}^{\beta_w - 1} \right) d\hat{\Phi} \\ &= \int \prod_{t=1}^{T} \left(\frac{1}{\Delta(\overrightarrow{\beta})} \prod_{w=1}^{W} \varphi_{t,w}^{\beta_w + n_{t,w} - 1} d\Phi_t \right) \\ &= \prod_{t=1}^{T} \frac{\Delta(\overrightarrow{\beta} + \overrightarrow{n_t})}{\Delta(\overrightarrow{\beta})} \end{aligned} \tag{9}$$

$$P(\overrightarrow{w^{(-i)}}|\overrightarrow{t^{(-i)}}) = \prod_{t=1}^{T} \frac{\Delta(\overrightarrow{\beta} + \overrightarrow{n_t^{(-i)}})}{\Delta(\overrightarrow{\beta})} \tag{10}$$

where matrix $\hat{\Phi} = \{\Phi_t\}_{t=1}^{T}$, $n_{t,w}$ is the number of times topic t is to be assigned to w in $\overrightarrow{w}$, $n_{t,w}^{(-i)}$ is the number

of times topic t is to be assigned to w in $\overrightarrow{w^{(-i)}}$, $\overrightarrow{n_t} = \{n_{t,w}\}_{w=1}^{W}$, and $\overrightarrow{n_t^{-i}} = \{n_{t,w}^{-i}\}_{w=1}^{W}$

Therefore,

$$\frac{P(\overrightarrow{w}|\overrightarrow{t})}{P(\overrightarrow{w^{(-i)}}|\overrightarrow{t^{(-i)}})} = \prod_{t=1}^{T} \frac{\Delta(\overrightarrow{\beta} + \overrightarrow{n_t})}{\Delta(\overrightarrow{\beta} + \overrightarrow{n_t^{(-i)}})}$$
$$= \frac{\Delta(\overrightarrow{\beta} + \overrightarrow{n_{t_i}})}{\Delta(\overrightarrow{\beta} + \overrightarrow{n_{t_i}^{(-i)}})}$$
$$= \frac{\beta_{w_i} + n_{t_i,w_i}^{(-i)}}{\sum_w (\beta_w + n_{t_i,w}^{(-i)})} \tag{11}$$

Finally, we arrive at:

$$P(t_i|\overrightarrow{w}, \overrightarrow{t^{(-i)}}) \propto P(t_i, w_i|\overrightarrow{w^{(-i)}}, \overrightarrow{t^{(-i)}})$$
$$= \frac{\alpha_{\mathcal{D},t_i} + n_{\mathcal{D},t_i}^{(-i)}}{\sum_t (\alpha_{\mathcal{D},t} + n_{\mathcal{D},t}^{(-i)})} \cdot \frac{\beta_{w_i} + n_{t_i,w_i}^{(-i)}}{\sum_w (\beta_w + n_{t_i,w}^{(-i)})}$$
$$= \begin{cases} \frac{n_{\mathcal{D},t_i}^{(-i)} + k\alpha}{(\sum_{t'=1}^{T} n_{\mathcal{D},t'}^{(-i)}) + (k+T-1)\alpha} \frac{n_{t_i,w_i}^{(-i)} + \beta_{w_i}}{\sum_{w'=1}^{W} (n_{t_i,w'}^{(-i)} + \beta_{w'})}, \\ \quad \text{if } t_i = \mathcal{D} \\ \frac{n_{\mathcal{D},t_i}^{(-i)} + \alpha}{(\sum_{t'=1}^{T} n_{\mathcal{D},t'}^{(-i)}) + (k+T-1)\alpha} \frac{n_{t_i,w_i}^{(-i)} + \beta_{w_i}}{\sum_{w'=1}^{W} (n_{t_i,w'}^{(-i)} + \beta_{w'})}, \\ \quad \text{if } t_i \neq \mathcal{D} \end{cases} \tag{12}$$

The Gibbs sampling algorithm to assign topic IDs to words is summarized as follows.

(1) **Initial step**

For a directory $\mathcal{D}$, we assign each word w in that directory an arbitrary topic ID t by using a distribution biased to directory $\mathcal{D}$: $(p_1 = \frac{1}{k+T-1}, ..., p_{\mathcal{D}} = \frac{k}{k+T-1}, ..., p_T = \frac{1}{k+T-1})$.

(2) **Update step**

For each word w_i, we randomly reassign its topic according to the distribution in Eq. (12).

(3) **Repeat update step until convergence**

Assume that we want to generate a pair of topic and word (t^*, w^*) for a new slot in directory $\mathcal{D}$. We can use Eq. (5) to model the probability of generating (t^*, w^*), where the first and second terms can be used to model the generation process of word w^* from topic $t*$ and the generation process of topic t^*, respectively. Therefore, parameter vectors $\Theta_{\mathcal{D}}$ and Φ_t can be derived from the topic IDs of words as follows.

$$\vartheta_{\mathcal{D},t^*} = P(t^*|\overrightarrow{t}) = \frac{P(\overrightarrow{t^*})}{P(\overrightarrow{t})}$$
$$= \begin{cases} \frac{n_{\mathcal{D},t^*} + k\alpha}{(\sum_{t'=1}^{T} n_{\mathcal{D},t'}) + (k+T-1)\alpha}, \text{if } t^* = \mathcal{D} \\ \frac{n_{\mathcal{D},t^*} + \alpha}{(\sum_{t'=1}^{T} n_{\mathcal{D},t'}) + (k+T-1)\alpha}, \text{if } t^* \neq \mathcal{D} \end{cases} \tag{13}$$

$$\varphi_{t^*,w^*} = P(w^*|\overrightarrow{w}, \overrightarrow{t^*})$$
$$= P(w^*|\overrightarrow{w}, t^*, \overrightarrow{t})$$
$$= \frac{P(w^*|\overrightarrow{w}, t^*, \overrightarrow{t}) P(\overrightarrow{w}|t^*, \overrightarrow{t})}{P(\overrightarrow{w}|t^*, \overrightarrow{t})}$$
$$= \frac{P(w^*, \overrightarrow{w}|t^*, \overrightarrow{t})}{P(\overrightarrow{w}|t^*, \overrightarrow{t})}$$
$$= \frac{P(\overrightarrow{w^*}|\overrightarrow{t^*})}{P(\overrightarrow{w}|t^*, \overrightarrow{t})}$$
$$= \frac{P(\overrightarrow{w^*}|\overrightarrow{t^*})}{P(\overrightarrow{w}|\overrightarrow{t})}$$
$$= \frac{n_{t^*,w^*} + \beta_{w^*}}{\sum_{w'=1}^{W} (n_{t^*,w'} + \beta_{w'})} \tag{14}$$

Here, we have $P(\overrightarrow{w}|t^*, \overrightarrow{t}) = P(\overrightarrow{w}|\overrightarrow{t})$, since $\overrightarrow{w}$ does not depend on t^*.

3.3 Modification of documents by extracted topics

We use topics extracted from web directories to model latent topics in new web documents and modify new documents as follows. We model the topic distribution of a web document by associating a distribution with each word in the document. We use the word distributions Φ_i of the extracted topics to model the distribution for each word. We then update the topic distributions of the words in a document by using an algorithm that is similar to the Gibbs sampling algorithm. The details are as follows.

(1) **Initial step**

The topic distribution of word w from web directories can be calculated as follows.

$$P(t_w = t|w) = \frac{P(t,w)}{P(w)} = \frac{P(t)P(w|t)}{P(w)}$$
$$\propto P(t)P(w|t) = P(t)\varphi_{t,w} \tag{15}$$

In Eq. (15), $P(w)$ is the same for all t, $P(t)$ is proportional to the number of word slots assigned to topic t in the learning phase, and $P(w|t) = \varphi_{t,w}$ is an output parameter in the learning phase.

(2) **Update step**

As can be seen from Eqs. (13) and (14), the first factor in Eq. (12) is equivalent to $P(t|d)$ and the second factor is equivalent to $P(w_i|t)$. Therefore, the update step of the Gibbs sampling algorithm can be rewritten as follows.

$$P_{new}(t|w_i) \propto P(t|d)P(w_i|t) \tag{16}$$
$$P(t|d) \propto \sum_{w \in d} P(t|w) \tag{17}$$

We update the topic distributions of words in a sim-

ilar manner as follows.

$$P_{new}(t|w_i) = \frac{P(t|d)P(w_i|t)}{\sum_t P(t|d)P(w_i|t)} \quad (18)$$

$$P(t|d) = \frac{\sum_{w \in d} P(t|w)}{\sum_t \sum_{w \in d} P(t|w)} \quad (19)$$

$$P_{update}(t|w_i) = \gamma P_{old}(t|w_i) + (1-\gamma)P_{new}(t|w_i) \quad (20)$$

Here, we used a smoothing technique while updating the words' topic distributions. In the experiment, we updated the topic distributions of words in a document 100 times and used a smoothing factor of $\gamma = 0.95$.

3.4 Measurement of document similarities

By associating a topic distribution with each word in document, we can consider an appearance of w as an appearance of T words $w^{(1)}, w^{(2)}, ..., w^{(T)}$, where each $w^{(i)}$ has a weight $P(t_i|w)$. Accordingly, the original $d = (w_1, w_2, ..., w_l)$ becomes $d^{(T)} = (w_1^{(1)}, w_1^{(2)}, \cdots, w_1^{(T)}, w_2^{(1)}, w_2^{(2)}, \cdots, w_2^{(T)}, \cdots, w_l^{(1)}, w_l^{(2)}, \cdots, w_l^{(T)})$. We represent $d^{(T)}$ by an extended topic-word vector as follows.

$$entropy(w) = \log T + \sum_{t=1}^{T} P(t|w) \log P(t|w) \quad (21)$$

$$weight(w,t) = entropy(w)P(t|w) \quad (22)$$

$$\overrightarrow{d^{(T)}} = \Big(weight(w_1,t_1), weight(w_1,t_2), \cdots, weight(w_1,t_T), weight(w_2,t_1), weight(w_2,t_2), \cdots, weight(w_2,t_T), \cdots weight(w_l,t_1), weight(w_l,t_2), \cdots, weight(w_l,t_T)\Big) \quad (23)$$

Using this $\overrightarrow{d^{(T)}}$, we can redefine the document similarity calculations in a tf-idf vector space model as follows.

$$Sim(d_1^{(T)}, d_2^{(T)}) = \overrightarrow{d_1^{(T)}}\overrightarrow{d_2^{(T)}} \quad (24)$$

Here, $P(t|w_{1,i})$ acts as frequency tf and $weight(w)$ acts as informativeness idf in the traditional tf-idf vector space model [1]. The meaning of Eq. (21) can be explained as follows. Given the fact that w has been observed, we can find the topic distribution w: $(p(t_1|w), p(t_2|w), ..., p(t_T|w))$. If w has not been observed, the topic distribution is the same for all topics: $(\frac{1}{T}, \frac{1}{T}, ..., \frac{1}{T})$. Therefore, the information amount conveyed by w is the difference in information amount between these two topic distributions, which is Eq. (21).

Table 1 List of 24 name queries

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

4 Data sets

4.1 Data sets of ambiguous names

The experiments used two data sets of ambiguous names.

Our own data sets

We put 24 personal names (see Table 1) in queries to the Google search engine[1)] and got the top 100 results for each query. These names were of researchers in computer, physics, medicine, and history. There were about 70 to 80 documents per name after we removed the noisy documents. Each data set contained documents mentioning researchers listed in Table 1 as well as documents mentioning other people.

In order to get a large number of data sets automatically, we mixed pairs of search result sets to create a pseudo namesake dataset. We mixed two results sets of two name queries. Then, we replaced all personal names in documents by a common name X to create a set of documents with pseudo ambiguous names. We only selected two name queries from different research fields, so that ambiguously named people in the mixed set had different professional careers. The experimental sets created in this way contained documents that mentioned to two selected researchers and other ambiguously named people. Since we had four research fields and six names per field, the number of data sets created in this way was $\binom{4}{2} \times 6 \times 6 = 216$.

Web People Search data sets

Besides the above data sets, we also carried out experiments on objective data sets created by other people, i.e., data sets from the Web People Search (WePS) task[2)] at SemEval2007[3)]. The WePS collection contained a training part of 49 data sets and a test part of 30

[1)] http://www.google.com
[2)] http://nlp.uned.es/weps/
[3)] http://nlp.cs.swarthmore.edu/semeval/index.php

Table 2 WePS dataset

Data set	Number of names	Average entities per name	Average document per name
Training	49	10.76	71.02
Test	30	45.93	98.93

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

data sets. These data sets were search results for names of people mentioned in Wikipedia, the ECDL06 conference, the ACL06 conference, and US Census data (see Table 2).

4.2 Data set of web directories

We selected web directories from three well-known collections of web directories: the Dmoz collection[4], the Google collection[5], and the Yahoo collection[6]. The directories were organized hierarchically. We selected the directories to use in our experiments as follows. First, we selected all level-two child nodes, starting from the root node in a collection. Then, we removed directories that had few documents, since topics might not be strong in such directories. We used two thresholds (10 and 20) as the floor number of documents and got two directory sets from each directory collection (six sets in total). Table 3 lists the number of directories and documents in each set.

We used the LDA algorithm described in section 3.2 to extract latent topics from each directory set. The parameters for LDA were set as follows; the number of topics was chosen to be equal to the number of web directories and the bias factor was 1, 10, 20, 50, 100, or 200. For the vocabulary of terms, we removed terms with frequencies less than 10 and got a vocabulary of roughly 10000 terms.

[4] http://www.dmoz.org
[5] http://directory.google.com
[6] http://dir.yahoo.com

5 Experiments

5.1 Experiment procedures

Document processing

We processed the documents and disambiguated the names appearing in them as follows.

1. We removed stop words and stemmed words to their root forms. We selected words surrounding personal names with a window of 100 to create a bag of words vector for each document.
2. We modified the document vectors using the method described in Section 3.3 and measured the document similarities.
3. We re-ranked documents according to the similarity values and evaluated the disambiguation performance. The end of this section describes the method of name disambiguation by re-ranking documents in more detail.

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

Vector Space Model method

For VSM, we removed stop words, stemmed words, and created bags of words for the documents. We used the tf-idf model to build document vectors and calculated the inner products as the document similarities. Re-ranking and disambiguation were done the same way as in our method.

Named Entity Recognition method

For NER, we used the Lingpipe tool[7] to extract named entities and built a bag of entity names for each document. We calculated the inner products as the document similarities and re-ranked documents to disambiguate names.

Name disambiguation by re-ranking documents

The research on word sense disambiguation and the previous research on personal name disambiguation often used clustering methods to disambiguate the different senses of a word. Instead, we used the re-ranking method to disambiguate names for the following reasons. When searching for people, users are likely interested in only one person; the clustering method might not exactly suit this interest. Furthermore, if the clustering results have mistakes, users have to check every cluster to look for correct answers; thus, clustering would not be cost effective for our purpose. Re-ranking documents to bring useful documents to the top might meet the users' requests precisely. Our system interacts with users in order to understand their requests better. Users select a useful document and show it to the system. The system re-ranks the rest of the documents according to their similarity to the selected document.

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

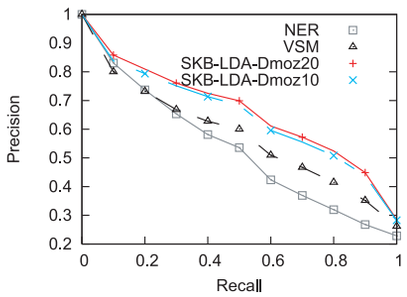

Fig. 2 Performance of SKB-LDA with Dmoz directories.

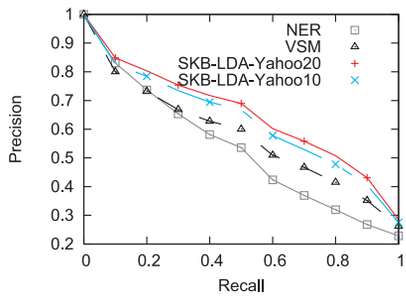

Fig. 3 Performance of SKB-LDA with Yahoo directories.

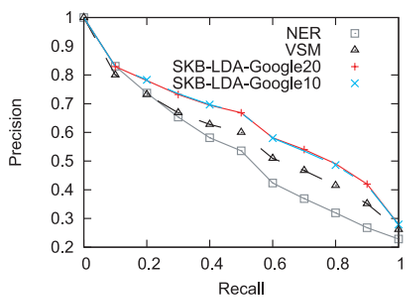

Fig. 4 Performance of SKB-LDA with Google directories.

5.2 Evaluation metrics

We evaluated the re-ranking performance as follows. For each document in an ambiguous-name document set, we evaluated the re-ranking precision for that document. We recorded precision values at 11 recall points: $P(d, 0\%)$, $P(d, 10\%)$,...,$P(d, 100\%)$. Then we evaluated re-ranking performance for each test set by taking the average re-ranking precision across all documents in the test set. For the test document set D, we calculated the average precision values at 11 recall points.

$$P(D, k\%) = \frac{\sum_{d \in D} P(d, k\%)}{|D|} \tag{25}$$

where $|D|$ is the number of documents in D, and $k = 0, 10, ..., 100$. Finally, we took the average precisions across all test sets.

$$P(k\%) = \frac{\sum_{D} P(D, k\%)}{N} \tag{26}$$

where N is the number of test sets.

5.3 Experiment results

Results with pseudo namesake data sets

We carried out experiments with our own pseudo ambiguous name data sets. We re-ranked documents using document similarities results calculated by SKB-LDA and baseline methods and then evaluated the performance of each method using Eqs. (25) and (26). The results shown in Figs. 2, 3, 4, and Table 4 prove that our method improves performance by 6% to 20% compared with baseline methods.

We assessed the SKB-LDA method with different bias factors. Tables 5 and 6 show the results for which we were able to verify the effectiveness of the bias factors. In particular, performance improved when the bias factor was from 20 to 100.

Results for WePS data sets

The experiments with WePS data sets compared our method with a baseline. Figure 6 and Table 7 show the results and they prove that our method improves performance by 4% to 5% compared with baseline methods.

Table 4 Performances of VSM, NER and SKB-LDA with pseudo ambiguous names.

Method	P_{aver}
VSM	58.5%
NER	54.1%
SKB_LDA_Yahoo10	61.9%
SKB_LDA_Yahoo20	64.0%
SKB_LDA_Google10	64.9%
SKB_LDA_Google20	65.9%
SKB_LDA_Dmoz10	65.8%
SKB_LDA_Dmoz20	**67.6%**

Table 5 Performance of SKB-LDA with different bias factors (1).

Bias	Google20	Yahoo20	Dmoz20
1	64.27%	64.02%	64.55%
10	63.63%	63.27%	64.90%
20	64.29%	65.15%	65.35%
50	63.79%	**65.42%**	66.31%
100	65.02%	65.20%	65.40%
200	**66.26%**	62.65%	**66.56%**

Table 6 Performance of SKB-LDA with different bias factors (2).

Bias	Google10	Yahoo10	Dmoz10
1	63.71%	62.90%	65.24%
10	64.12%	64.05%	65.21%
20	**64.51%**	**64.54%**	65.31%
50	63.71%	63.40%	65.09%
100	64.15%	64.17%	**65.46%**
200	63.42%	64.40%	65.12%

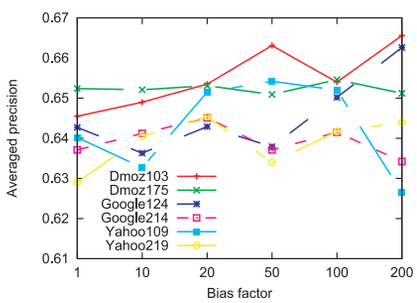

Fig. 5 Performance of SKB-LDA with different bias factors.

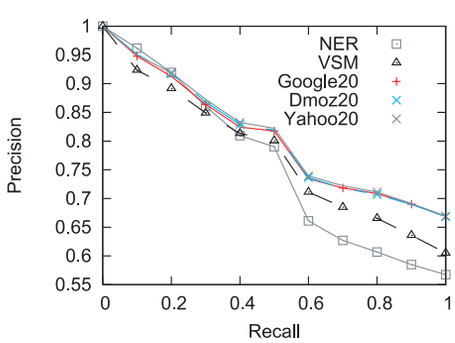

Fig. 6 Comparison of performances by approaches using WePS data set.

Table 7 Performances with WePS dataset.

Method	Averaged precision
NER	76.26%
VSM	78.01%
SKB-LDA Google20	**80.79%**
SKB-LDA Dmoz20	**80.94%**
SKB-LDA Yahoo20	**81.17%**

6 Discussion

Here, we discuss the advantages and disadvantages of using external knowledge in personal name disambiguation.

The main advantage of using web directories as external knowledge is that we can exploit the richness of information in web directories to complement to sparse information in web documents. The relevant text in web documents may be short and contain noise, so important terms may not appear frequently. This makes it difficult to extract important contexts and degrades disambiguation performance. By using web directories, we can find important terms that appear frequently in web directories, and we can use the occurrences of important terms in web directories to support recognition of important terms in web documents. We can use topics from web directories to modify web documents so that document topics will be closer to the topics in the web directories. This helps to increase the weight of important terms in web documents and improves disambiguation performance.

Another advantage of using web directories is that cost of preparation is low but topic coverage is wide. Well-prepared web directory collections already exist, and we can directly reuse them without going to much labor. Web directories also contain a large variety of topics so that they can work for people with different careers and different contexts.

Our approach has a disadvantage in that it requires more computation costs for document similarity computations. The costs can be divided into offline calculation costs and online calculation costs. Offline calculation costs are those of extracting latent topics from web directories. These calculations can be done in advance and do not affect the response time of the search system. Online calculation costs are those of the document vector modifications and document similarity calculations. These calculations are carried out upon the user's request, and they lengthen the search system's response time. In our experiments, this online calculation time was about ten times longer than that for the VSM method.

Web directories can be utilized in a different way for the name disambiguation problem. For example, we can regard these directories as a set of text categories to build a text categorization with text mining techniques. This text categorization can be used to calculate topic feature vectors for documents and to measure document similarities from these vectors. We will investigate this approach carefully in our future work.

7 Conclusions

The need for searching for personal information on the WWW is growing. We tried to solve the name disambiguation problem and developed a new method to improve disambiguation performance. The new method involves extracting topics from web directories

and using these topics to complement contexts in search result documents. In this way, it enables important information about people to be more easily recognized. The document similarities are calculated and documents are re-ranked to discriminate ambiguous names. Experimental results showed that the use of web directories improves disambiguation performance. In the future, we will combine our method with other methods such as NER and keyword extraction so that we can exploit more useful information and disambiguate names better.

References

[1] A. Aizawa, "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, New York, NY, USA, 2000. ACM Press.

[2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison Wesley Longman Publishing, 1999.

[3] A. Bagga and B. Baldwin, "Entity-based cross-document coreferencing using the vector space model", In *ACL1998*, 1998.

[4] R. Bekkerman and A. McCallum, "Disambiguating web appearances of people in a social network", In *The Fourteenth International World Wide Web Conference, WWW2005*, 2005.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation", *J. Mach. Learn. Res.*, vol.3, pp.993–1022, 2003.

[6] D. Bollegala, Y. Matsuo, and M. Ishizuka, "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*, 2006.

[7] K. T. Frantzi, S. Ananiadou, and J. Tsujii, "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, London, UK, 1998. Springer-Verlag.

[8] T. L. Griffiths and M. Steyvers, "Finding scientific topics", *Proc Natl Acad Sci U S A*, 101 Suppl 1, pp.5228–5235, April 2004.

[9] B. Malin, "Unsupervised name disambiguation via social network similarity", In *Workshop on Link Analysis, Counterterrorism, and Security, SIAM2005*, 2005.

[10] G. S. Mann and D. Yarowsky, "Unsupervised personal name disambiguation", In *Proceedings of Computational Natural Language Learning 2003*, 2003.

[11] T. Pedersen, A. Kulkarni, R. Angheluta, Z. Kozareva, and T. Solorio, "Name discrimination by clustering similar contexts", In *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics*, 2005.

[12] D. Ravichandran and E. Hovy, "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, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[13] H. Schutze, "Automatic word sense discrimination", *Computational Linguistics*, vol.24, no.1, pp.97–123, 1998.

[14] Q. M. Vu, T. Masada, A. Takasu, and J. Adachi, "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, New York, NY, USA, 2007. ACM Press.

[15] Q. M. Vu, T. Masada, A. Takasu, and J. Adachi, "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, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[16] Q. M. Vu, A. Takasu, and J. Adachi, "Improving the performance of personal name disambiguation using web directories", *Inf. Process. Manage.*, vol.44, no.4, pp.1546–1561, 2008.

[17] X. Wan, J. Gao, M. Li, and B. Ding, "Person resolution in person search results: Webhawk", In *Proceedings of the ACM Fourteenth Conference on Information and Knowledge Management, CIKM2005*, 2005.

Quang Minh VU

Quang Minh VU received his B.E. from Kyoto University in 2003, his M.E. from the University of Tokyo in 2005 and his Ph.D from the University of Tokyo in 2008. He has done researches in several research areas in computer science including computer hardware optimization, network architecture design, and text mining. He is now a postdoc researcher at National Institute of Informatics, Japan, doing a research on a management system of bibliography database.

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 Professor in the Digital Content and Media Sciences Research Division, National Institute of Informatics (NII), Japan. He is also the Director of the Cyber Sceince Infrastructure Development Department of NII. His professional career has largely been spent in research and development of scholaly information systems, such as NACSIS-CAT and NII-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 his BE, ME and Doctor of Engineering in Electrical Engineering from the University of Tokyo in 1976, 1978, and 1981, respectively. He is a member of IPSJ, IEEE, and ACM.