Name Disambiguation Boosted by Latent Topics from Web Directories

Quang Minh Vu Atsuhiro Takasu Jun Adachi
National Institute of Informatics, Tokyo, 101-8430 Japan
{vuminh, takasu, adachi}@nii.ac.jp

Abstract

Search results for personal name queries often contain documents relevant to several people as a personal name is often shared by several people. In order to differentiate people in these search results, it is required to extract contexts relevant to people in documents. However, since web documents are noisy and the texts related to people might be short, it is difficult to extract contexts of people effectively. We propose a new method that uses web directories as additional information in order to recognize topic terms in documents more easily and to extract contexts of people more effectively. First, we apply latent Dirichlet allocation method to extract latent topics in web directories. Then, the extracted topics are used to recognize topics contained in name ambiguity documents so that common context measurements can be calculated more effectively. Our experiments, conducted with documents of real people in the web and several well-known web directories, show that our approach disambiguates personal names better than some other conventional approaches like vector space model approach and named entity recognition approach.

1. Introduction

Information related to people plays an important part in the World Wide Web (WWW). A certain portion of search queries is to search for people using their names[6]. Since search engines use only name queries to filter documents, the results often contain documents referring to several people sharing the same name. Therefore, users have to manually select the person of interests from the results. In our research, we have used a re-ranking method to help users find the correct documents more quickly. First, users select from the result set a document that mentions the person of interests. Then, the system re-ranks the documents in the result set in the order of relevance to the selected document, so that useful documents go to the top.

In order to create re-ranked results of high quality, it is very crucial to measure similarities of document pairs correctly. In previous researches, several approaches have been proposed to measure document similarities. They are vector space model method (VSM)[2], named entity recognition method (NER)[7, 14], and context extraction method[9]. However, these methods are limited when applying to web documents that mention general people. Common context measurement of these documents is more challenging because the set of documents relevant to an arbitrary person might be small. Therefore, frequencies of important words are few and their weight differ only a little from weight features of other words.

In our research, we use topics extracted from directories such as the Dmoz directory[1], the Yahoo directory[2] and the Google directory[3] to complement the contexts of people that appear in web documents. In [12, 13], we proposed some heuristic methods to utilize the frequencies of important words in web directories to modify the weights of important words in documents. In this paper, we apply latent Dirichlet allocation method (LDA)[3] to obtain a probabilistic latent topic model for web directories and then to extract features of online web documents using the obtained model. Using topics' word distributions, we modify documents so that important words will have more weights. Then, we calculate the similarities of modified documents to measure the common contexts in original documents.

The main contributions of this paper to the personal name disambiguation problem are as follows.

1. Utilization of web directories to disambiguate personal names more effectively. In web directories,

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

important words are supposed to appear more frequently because web directories have much more amount of texts. Therefore, we use web directories to recognize important words in other web documents more easily.

2. Application of LDA to the name disambiguation problem using web directories. In the conventional LDA method, a document receives influences from all topics equally. However, since documents in the same directory are close in topic, we assume that each directory receives more influences from one specific topic and equal influences from the remaining topics. We model this assumption by using a biased hyper-parameter for one specific topic of each directory.

3. Latent topic based document similarity. We use the topics extracted from web directories to calculate topic distributions of terms in online documents and use these topic distributions to improve document similarity calculations.

The rest of this paper is organized as follows. In Section 2, we summarize previous works on document similarity measurements and name disambiguations. In Section 3, we introduce our approach in details. Then, in Section 4, we report our experiments that try to disambiguate personal names in real web documents using additional information from web directories. We discuss the advantages and disadvantages of our approach and compare it with other approaches in Section 5. Finally, we conclude our research in Section 6.

2. Related works

In [2], the authors used vector space model method (VSM) to create feature vectors for documents and used inner products of feature vectors to measure document similarities. This approach performs well with name disambiguation of people mentioned in newspapers. In [9], the authors applied the singular value decomposition method[8] for a term co-occurrence matrix to build contexts for terms and then they summarized the contexts of terms in a document to build a context for that document[11]. This approach works well when there were a large number of documents related to one person. It was applied to disambiguate famous people mentioned in large numbers of web documents. Additional information resources were also used to improve the disambiguation performances. In [6], the authors used some databases as additional information. They used the names of book authors from online bookstores and the research keywords in the DBLP database[4] to extract relevant information more easily. Named entity recognition (NER) method[10] used in [7, 14] could also be regarded as a method that used additional information resources, since training data were used to train the NER algorithm.

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

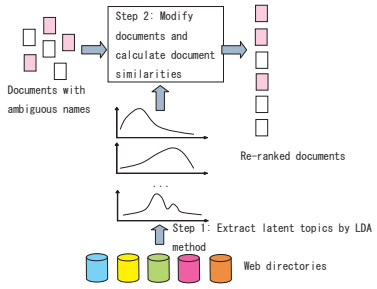

Figure 1. Overview of our approach

Name disambiguation in searching for general people on the web is a new problem that has not been targeted so much in previous researches. This problem has the following characteristics. First, it differs from newspaper documents in that it is much noisy. Second, the amount of information relevant to one person varies across people. Famous people might have an abundance of relevant texts, while ordinary people might have only a few relevant texts. Due to these different characteristics, we cannot directly apply previous approaches to this new problem of name disambiguation for the following reasons. The VSM approach may not perform well with noisy documents. Noise also reduces the precision of the NER algorithm. Meanwhile, the context extraction approach is also limited when the number of documents is small.

3. Our approach

3.1 Overview

We propose to use web directories as a resource of additional knowledge for disambiguating personal names. Web directories contain many rich contexts that can be used to complement the contexts of people in web documents. Fig. 1 shows an overview of our approach, which consists of two steps. In the first step, we use LDA method to extract contexts in the form of topics and word distributions of topics. Then, in the second step, the extracted topics are used to complement contexts of people in web documents. The complementation is done by modifying documents to make measurement of common important contexts easier. We name our method **"Similarity via Knowledge Base using LDA method (SKB-LDA)"**. In the following Subsections, we will discuss about our approach in more details.

3.2 Extraction of topics in web directories

We modified LDA method[3, 5] to make it work well with web directories as follows. Since we know that documents in the same directory have the similar topic distributions, we use the same topic distribution for all documents in the same directory. We parameterize the topic distributions of directories and the word distributions of topics as follows.

Let D, T, and W be the number of web directories, the number of latent topics, and the number of different words, respectively. For a directory i, its topic distribution is represented by a vector $\Theta_i = (\vartheta_{i,1}, \vartheta_{i,2}, ...\vartheta_{i,T})$. In the conventional LDA method[3], the distribution density of these vectors is assumed to follow the same Dirichlet distribution for all documents. However, since we know that each directory is strongly related to its own specific topic, we assume that the documents in a directory are influenced mainly by the specific topic associated with that directory, while receiving small influences from topics of other directories. We model this assumption by using different sets of hyper-parameters to build different Dirichlet distributions for different directories. A hyper-parameter vector $\vec{\alpha}^{(i)}$ for a directory i is set to have a large hyper-parameter $\alpha_i = k\alpha$ for its associated specific topic, while having small hyper-parameters $\alpha_j = \alpha$ for other topics $j \neq i$ as follows

$$\vec{\alpha}^{(i)} = (\alpha, \alpha, ..., \alpha_i = k\alpha, ..., \alpha). \tag{1}$$

We call k the bias factor of directories. The topic vector of directory i has a distribution density that follows a Dirichlet distribution $\mathcal{DIR}(\vec{\alpha}^{(i)})$.

For a topic t, its word distribution is represented by a vector $\Phi_t = (\varphi_{t,1}, \varphi_{t,2}, ...\varphi_{t,W})$. The distribution density of Φ_t is also assumed to follow a Dirichlet distribution $\mathcal{DIR}(\vec{\beta})$, where $\vec{\beta} = (\beta_1, \beta_2, ..., \beta_W)$ and $\beta_1 = \beta_2 = \cdots = \beta_W$ are hyper-parameters.

In order to find the latent topics and the distributions of topics for documents, we use the Gibbs sampling algorithm to assign topic IDs to words in documents and infer the parameter vectors Θ and Φ from words' topic IDs[5]. Denote $\overrightarrow{w} = (w_1, w_2, ..., w_L)$ as the vector composed by lining up all words in all documents, and denote t_i as the topic ID assigned to word w_i. The procedure of the Gibbs sampling algorithm for LDA method with hyper-parameters biased to directories is as follows.

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

2. **Update step**
   Let $\overrightarrow{t_{-i}}$ denote $(t_1, t_2, ...t_{i-1}, t_{i+1}, ...t_L)$. For each word w_i, we reassign its topic randomly according to the following distribution.

$$P(t_i = t|\overrightarrow{t_{-i}}, \overrightarrow{w}) \propto \begin{cases} \frac{n^{(t)}_{-i,dir_d}+k\alpha}{\left(\sum_{t'=1}^{T} n^{(t')}_{-i,dir_d}\right)+(k+T-1)\alpha} \frac{n^{(w)}_{-i,topic_t}+\beta_w}{\sum_{w'=1}^{W}\left(n^{(w')}_{-i,topic_t}+\beta_{w'}\right)}, \\ \quad \text{if } t = dir_d \\ \frac{n^{(t)}_{-i,dir_d}+\alpha}{\left(\sum_{t'=1}^{T} n^{(t')}_{-i,dir_d}\right)+(k+T-1)\alpha} \frac{n^{(w)}_{-i,topic_t}+\beta_w}{\sum_{w'=1}^{W}\left(n^{(w')}_{-i,topic_t}+\beta_{w'}\right)}, \\ \quad \text{if } t \neq dir_d \end{cases} \tag{2}$$

   where $w = w_i$, dir_d is the directory containing w_i, $n^{(t)}_{-i,dir_d}$ is the number of words other than w_i in dir_d to be assigned to topic t, and $n^{(w)}_{-i,topic_t}$ is the total number of times words w other than w_i are assigned topic t.

3. Repeat update step until convergence.

The parameter vectors Θ and Φ can be derived from topic IDs of words as follows

$$\vartheta_{dir,t} = \begin{cases} \frac{n^{(t)}_{dir_d}+k\alpha}{\left(\sum_{t'=1}^{T} n^{(t')}_{dir_d}\right)+(k+T-1)\alpha}, \text{if } dir = t \\ \frac{n^{(t)}_{dir_d}+\alpha}{\left(\sum_{t'=1}^{T} n^{(t')}_{dir_d}\right)+(k+T-1)\alpha}, \text{if } dir \neq t \end{cases} \tag{3}$$

$$\varphi_{t,w} = \frac{n^{(w)}_{topic_t} + \beta_w}{\sum_{w'=1}^{W}\left(n^{(w')}_{topic_t} + \beta_{w'}\right)}, \tag{4}$$

where $n^{(t)}_{dir_d}$ is the number of words in dir_d to be assigned the topic t and $n^{(w)}_{topic_t}$ is the total number of times the word w is assigned the topic t.

3.3 Document similarities

Using words' topic distributions to calculate document similarities

As we can see from Eq. (2), in the LDA method we can model a word w in a document d with a topic distribution. Using this topic distribution $\{P(t_1|w,d), P(t_2|w,d), \cdots, P(t_T|w,d)\}$, we can consider an appeareance of w as appeareances of T words $w^{(1)}, w^{(2)}, \cdots, w^{(T)}$, where each $w^{(i)}$ has a weight $P(t_i|w,d)$. Upon this consideration, an original $d = (w_1, w_2, ..., w_l)$ becomes $d^{(T)} = (w_1^{(1)}, w_1^{(2)}, \cdots, w_1^{(T)}, w_2^{(1)}, w_2^{(2)}, \cdots, w_2^{(T)}, \cdots, w_l^{(1)}, w_l^{(2)}, \cdots, w_l^{(T)})$. Then, using this $d^{(T)}$, we can redefine the document similarity calculation in the tf-idf VSM as follows.

$$Sim(d_1^{(T)}, d_2^{(T)}) = \sum_{w_{1,i}=w_{2,j}} \sum_t P(t|w_{1,i}, d_1) weight(w_{1,i}, d_1) P(t|w_{2,j}, d_2) weight(w_{2,j}, d_2), \quad (5)$$

$$weight(w, d) = \log T + \sum_{t=1}^{T} P(t|w, d) \log P(t|w, d). \quad (6)$$

Here, $P(t|w_{1,i}, d_1)$ is the topic probability of $w_{1,i}$ in document d_1 and $P(t|w)$ is topic probability of w in directories. They are defined in details in the next part in this Subsection. $P(t|w_{1,i}, d_1)$ acts as the frequency *tf* and $weight(w)$ acts as the informativeness *idf* in the traditional tf-idf VSM. The meaning of Eq. (6) can be explained as follows. Given that w is observed in d, we learn the topic distribution associated with w: $(p(t_1|w, d), \cdots, p(t_T|w, d))$. If we have not observed w, then the topic distribution is the same for all topics: $(\frac{1}{T}, \cdots, \frac{1}{T})$. Therefore, the amount of information conveyed by w is the difference of information amount between these two topic distributions, which is Eq. (6).

Inference of words' topic distributions in a new document

We apply the Gibbs sampling algorithm to a new document as follows. For a new document, we first initialize the topic distribution for each word by using the observed topic distributions in the web directories. Then, we update the topic distribution for each word by using the topic distributions of other words in the same document.

1. **Initial step**
   The topic distribution of a word w observed from web directories can be calculated as follows

$$P_{init}(t_w = t|w, d) = P(t|w) = \frac{P(t)P(w|t)}{P(w)} \propto P(t)P(w|t) = P(t)\varphi_{t,w}. \quad (7)$$

   Here, $P(t)$ is proportional to the number of word slots assigned to topic t in the learning phase and $P(w|t) = \varphi_{t,w}$. For a term t that does not appear in web directories, we cannot calculate its topic distribution and we ignore this kind of term in our calculations.

2. **Update step**
   As can be seen from Eqs. (3) and (4), in Eq. (2), the first factor 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 rewrite as follows

$$P_{new}(t|w_i, d) \propto P(t|d)P(w_i|t), \quad (8)$$

$$P(t|d) \propto \sum_{w \in d} P(t|w, d). \quad (9)$$

   We update the topic distributions of words in a similar manner and the detailed calculations are as follows

$$P_{new}(t|w_i, d) = \frac{P(t|d)P(w_i|t)}{\sum_t P(t|d)P(w_i|t)}, \quad (10)$$

$$P(t|d) = \frac{\sum_{w \in d} P(t|w)}{\sum_t \sum_{w \in d} P(t|w)}, \quad (11)$$

$$P_{update}(t|w_i, d) = \gamma P_{old}(t|w_i, d) + (1 - \gamma) P_{new}(t|w_i, d). \quad (12)$$

   Here, we use a smoothing technique while updating words' topic distributions. In our experiment, we update topic distributions of words 100 times and use a smoothing factor of $\gamma = 0.95$.

4. Experiments

4.1 Data sets

Data sets of pseudo-ambiguous names

We used 24 name queries to get documents from the Google search engine. They were names of researchers specializing in some fields as listed in Table 1. We sent each name to the Google search engine and took the top 100 results. Then, we artificially created name ambiguity documents to obtain a large number of test sets as follows. We selected two result sets corresponding to two names and mixed them together. Then, we replaced personal names in the documents with the name X to create a set of documents containing ambiguous names. The two name queries were selected from two researchers in different fields, so that people in the mixed data set had different professional careers. Since we used four research fields and six personal names from each research field, this method of mixing documents yielded $\binom{4}{2} \times 6 \times 6 = 216$ test sets.

Data sets of real ambiguous names

Besides our own prepared data set, we also carried out experiments with an objective data set. We used the data set from the Web People Search Task (WePS)[5] at the SemEval-2007 workshop[6]. This data set contains a training set of 49 ambiguous names and a test set of 30 ambiguous names. In our experiment, we use both of the two sets as test sets. The ambiguous names in WePS are from Wikipedia, the ECDL06 conference, the ACL06 conference, and US Census data. More details can be found in [1].

[5] http://nlp.uned.es/weps/

[6] http://nlp.cs.swarthmore.edu/semeval/index.php

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

Table 2. Numbers of directories and documents in the 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 for web directories

We chose three well-known web directories to use in our experiments: the Dmoz directory, the Google directory and the Yahoo directory. For each directory resource, we selected document sets in an objective fashion as follows. The document sets in a directory were organized in a tree structure. We only selected up to the level two child nodes starting from the root node, since deeper nodes contain only small number of documents and topics are not strong there. Among the selected child nodes, we also removed document sets that have a small number of documents. For each directory, we used 10 and 20 as the threshold number of documents to create two directory structures. The details of six resulting directory structures are listed in Table 2.

We applied the LDA method and the Gibbs sampling algorithm to each directory structure to extract topics. We removed stop words and ignored words that appeared in less than 10 documents and got a vocabulary of roughly 10000 words. We set the number of topics equal to the number of directories $T = D$, since we only want to extract directories' specific topics. The other hyper-parameters were selected as follows: $\alpha = \frac{50}{T}, \beta = \frac{200}{W}$, and $k = 1, 10, 20, 50, 100, 200$.

4.2 Baseline methods

We compared our approach with two baseline methods: a VSM method and a NER method.

Vector space model method

In the VSM method, we removed stop words and used the Porter stemming algorithm[7] to change words to their root forms. Then, we chose n words before and n words after every personal name to create a document's bag of words and used the tf-idf method[8] to measure term weights. The document similarities were calculated by inner products of document vectors.

Named entity recognition method

In the NER method, we used the LingPipe software[8] to extract entity names in documents. Then, the entity names were used to construct named entity vectors of documents. The document similarities were calculated by inner products of the named entity vectors.

4.3 Evaluation metrics

We evaluated the disambiguation performance by measuring re-ranking performances.

Re-ranking performance for each document

For each document d in a test set, we re-ranked the other documents in that test set in order of their similarities to d, so that documents relevant to the person mentioned in d tended to go to the top. We recorded precision values at 11 recall points: $P(d, 0\%)$, $P(d, 10\%)$,...,$P(d, 100\%)$. Finally, we calculated the average precision among all documents in one test set and then the average precision among all test sets as follows.

Average of performance among all documents in each test set

For a test document set D, we calculated the averaged precision values at 11 recall points.

$$P(D, k\%) = \frac{\sum_{d \in D} P(d, k\%)}{|D|}, \tag{13}$$

where $|D|$ was the number of documents in D, and $k = 0, 10, ..., 100$.

Average of performance among all test sets

We calculated the average precision over all test sets as follows.

$$P(k\%) = \frac{\sum_{D} P(D, k\%)}{N}, \tag{14}$$

where N was the number of test sets.

4.4 Experimental results

We conducted experiments with our 216 test sets of pseudo-ambiguous names and the WePS test sets. The six directory structures were used independently to modify documents. We chose a window size $n = 100$ and bias factors $k = 1, 10, 20, 50, 100, 200$.

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

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

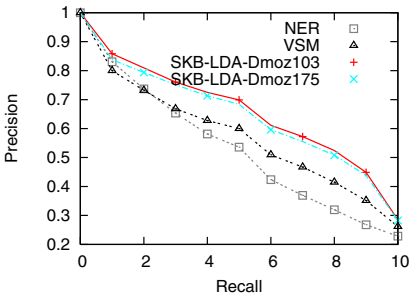

Figure 2. Performance of SKB-LDA with the Dmoz directories

Table 3. Performances of the baseline methods

Method	P_{aver}
Vector Space Model	58.5%
Named Entity Recognition	54.1%

The overall performance for each method

Table 3, 4, and 5 list the performances of the baseline methods and our method, respectively. Fig. 2 shows the precision-recall graphs for our method with the Dmoz directory structures and the comparisons with the baseline methods. In these results, our method outperformed both baseline methods, VSM and NER.

Performance of our approach when varying the bias factor

We varied the bias factor k used in the step of latent topic extraction, as described in Section 3.2 to compare our biased LDA model with normal LDA models. The values of the bias factor were $k = 1, 10, 20, 50, 100,$ and 200. When $k = 1$, our biased model becomes a normal LDA model, in which topics are symmetric to all directories. When $k > 1$, the topics are asymmetric and each directory is biased to its own specific topic. The performances with different bias factors in terms of the average precision are plotted in Fig. 3. As we can see from the graph, the effects of bias were significant for $k = 50, 100$ with the Dmoz103, Google124, and Yahoo109 directories.

Performances with real ambiguous names

The results with the WePS dataset are listed in Table 6. Our approach also outperformed VSM approach and NER approach in this experiment, as well.

Table 4. Performance of SKB-LDA with different bias factors

Bias	Google124	Yahoo109	Dmoz103
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 5. Performance of SKB-LDA with different bias factors

Bias	Google214	Yahoo219	Dmoz175
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%

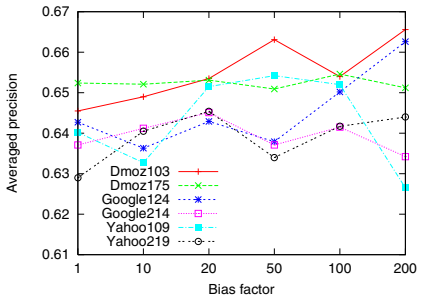

Figure 3. Performance of SKB-LDA with different bias factors

Table 6. Experimental results with WePS dataset

Method	Averaged precision
NER	76.26%
VSM	78.01%
SKB-LDA Google124	**80.79%**
SKB-LDA Dmoz103	**80.94%**
SKB-LDA Yahoo109	**81.17%**

5. Discussions

5.1 Comparison with VSM method

We have used web directories as additional information to improve the measurement of common contexts in documents. Some advantages of using web directories are as follows. Compared to a normal web page, web directories contain a greater abundance of text and their topics appear more strongly. We have used the LDA method to extract web directories' latent topics. Then, these extracted topics are combined with a Gibbs sampling calculation to analyze online documents' topics. This combination allows us to recognize topics that are strongly related to a word more easily. Upon recognization of words' important topics, we improve the traditional document similarity calculation in VSM, so that the meanings of words are considered more precisely. In our approach, a word's meaning is represented by a topic distribution for that word, and

this distribution can be observed more effectively with the use of web directories.

5.2 Comparison with NER method

In the WePS task at the SemEval-2007 workshop, the CU-COMSEM team[4], which achieved the best performance, used an NER technique and a co-reference technique to extract features of people. Their performance in terms of SemEval's $F_{measure}$ metric[1] has reached 78%. We applied our proposed similarity measurements to form clusters of people from WePS test sets using some clustering methods such as single-link, complete-link, and average-link agglomerative clustering. The resulting clustering performance was about 63%, which is lower than that of the CU-COMSEM team. Actually, the clustering task is difficult and most of the participating teams have the results ranging from 40% to 67%. In order to improve the performance of our method for the WePS task, we need to develop a clustering method suitable for our proposed similarity measurement. This is one of our future works.

Comparing to the method by CU-COMSEM team, our method has the following advantages. The NER technique and the co-reference technique require significant human effort to annotate the training data. They are difficult to apply to languages where natural language processing researche is not mature. In contrast to these techniques, our approach extracts latent topics in an unsupervised manner and it only requires a collection of documents on some topics. In our research, we have used web directories, but other kinds of resources like Wikipedia categories and newspaper archives could also be suitable.

Our approach has a disadvantage of increasing calculation complexity. In particular, the online computation time is also increased, since we have to modify online documents and documents' bags of words are T times longer than those in the VSM. In our experiments, the calculation time for our SKB-LDA method is about ten times longer than that for the VSM method.

6. Conclusions

We have proposed a new method that uses web directories as a kind of additional information to improve name disambiguation performances. We applied the LDA method to preprocess web directories and to extract the latent topics contained in the web directories. Then, the extracted topics are used to recognize online documents' topics and to disambiguate personal names in online documents. Our experimental results showed that the LDA method was effective in extracting latent topics and that the use of these topics improved the name disambiguation performance.

References

[1] J. Artiles, J. Gonzalo, and S. Sekine. The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 64–69, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

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

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

[4] Y. Chen and J. H. Martin. Cu-comsem: Exploring rich features for unsupervised web personal name disambiguation. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 125–128, Prague, Czech Republic, 2007. Association for Computational Linguistics.

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

[6] R. Guha and A. Garg. Disambiguating people in search. In *The Thirteenth International World Wide Web Conference, WWW2004*, 2004.

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

[8] C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 2003.

[9] 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.

[10] 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*, pages 41–47, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

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

[12] 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*, pages 839–843, New York, NY, USA, 2007. ACM Press.

[13] Q. M. Vu, A. Takasu, and J. Adachi. Improving the performance of personal name disambiguation using web directories. *Inf. Process. Manage.*, 44(4):1546–1561, 2008.

[14] 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.