Journal of Natural Language Processing Vol. 21 No. 4
translated from: Journal of Natural Language Processing Vol. 19, No. 3, pp. 121–142

Relevance Feedback using Surface and Latent Information in Texts

Jun Harashima[†] and Sadao Kurohashi[†]

Most relevance feedback methods re-rank search results using only the information of surface words in texts. We present a method that uses not only the information of surface words but also that of latent words that are inferred from texts. We infer latent word distribution in each document in the search results using latent Dirichlet allocation (LDA). When feedback is given, we also infer the latent word distribution in the feedback using LDA. We calculate the similarities between the user feedback and each document in the search results using both the surface and latent word distributions and re-rank the search results on the basis of the similarities. Evaluation results show that when user feedback consisting of two documents ($3,589$ words) is given, the proposed method improves the initial search results by 27.6% in precision at 10 (P@10). Additionally, it proves that the proposed method can perform well even when only a small amount of user feedback is available. For example, an improvement of 5.3% in P@10 was achieved when user feedback constituted only 57 words.

Key Words: *Information Retrieval, Relevance Feedback, Latent Dirichlet Allocation*

1 Introduction

The main purpose of information retrieval (IR) is to rank documents so that users can obtain information efficiently. However, appropriate ranking of documents is difficult to achieve through one-off retrievals because user queries are typically short and ambiguous (Jansen, Spink, and Saracevic 2000). For example, the query "Mac price" can be interpreted as the price of a "Mac" (PC), a food item at "McDonalds," or some other "Mac." If we do not know what "Mac" refers to, we cannot rank the documents in a way that satisfies user information needs.

Relevance feedback (RF) is a technique that solves this problem by incorporating user feedback into the IR process. The basic procedure for RF is as follows.

(1) The RF-based system presents the initial search results for a given query.

(2) The user selects some relevant documents from the search results.

(3) The system modifies the search results using this feedback.

[†] Graduate School of Informatics, Kyoto University

[*] This article has been partially revised for better understanding of overseas readers.

For example, if the system obtains documents about the price of a "Mac" (PC) as user feedback, it assumes that the user is interested in this topic and modifies the initial search results.

There are a variety of RF methods based on different retrieval models. Rocchio's algorithm (Rocchio 1971) and the Ide dec-hi method (Ide 1971) are well-known RF methods for the vector space model (Salton, Wong, and Yang 1975). In the probabilistic model (Spärck Jones, Walker, and Robertson 2000), feedback can be used to modify the weight of terms to change or expand the original query. For language modeling approaches (Ponte and Croft 1998), Zhai and Lafferty (2001) proposed a fundamental RF method.

The basic idea behind these methods is the same, i.e., documents that are similar to the feedback are re-ranked higher. It should be noted that most methods calculate similarities using information from words that only appear in the feedback and search results. In other words, these methods do not use information from words that do not appear in the given texts.

However, information from highly probable relevant words can be useful for re-ranking search results even if the words do not appear in the given texts. Consider the query, "Mac price," and suppose that the feedback contains documents about the price of a "Mac" (PC). Although the feedback may not contain words such as "CPU" and "HDD," these words are closely related to the feedback and are, therefore, highly probable. The same is true of other relevant documents in the search results. Even if a relevant document does not contain words specific to the feedback and if it does include closely related to words, these words are highly probable. This information can be useful for calculating similarities between the feedback and other documents.

In this paper, we propose an RF method that uses the surface information in texts and latent information contained in the texts. For each document in the search results, we infer the distribution of words that are highly probable given the latent topics in the document using latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003). We calculate the similarities between the feedback and each document in the search results using both surface and latent word distributions. Then, we re-rank the search results on the basis of the similarities.

The remainder of this paper is organized as follows. In Sections 2 and 3, we explain the language modeling approaches for IR and LDA, which form the basis of the proposed method. In Section 4, we present the proposed method. Section 5 reports experiments performed to evaluate the proposed method, and conclusions are presented in Section 6.

2 Language Modeling Approaches to IR

In this section, we describe the language modeling approaches for IR that form the basis of our method.

2.1 Overview

Language modeling approaches can be classified into three types: query likelihood model (Ponte and Croft 1998), document likelihood model (Lavrenko and Croft 2001), and Kullback-Leibler (KL) divergence retrieval model (Lafferty and Zhai 2001).

In the query likelihood model, a document language model $P_{\boldsymbol{d}_h}(\cdot)$ is constructed for each document $\boldsymbol{d}_h$ $(h = 1, \ldots, H)$ in the collection. When a query $\boldsymbol{q}$ is submitted by a user, the query likelihood $P_{\boldsymbol{d}_h}(\boldsymbol{q})$ is computed using the document model for each $\boldsymbol{d}_h$. Then, the documents in the collection are ranked according to their likelihood.

In the document likelihood model, a query language model $P_{\boldsymbol{q}}(\cdot)$ is constructed for a given query. The query language model is then used to compute the document likelihood $P_{\boldsymbol{q}}(\boldsymbol{d}_h)$ for each document in the collection. The documents are then ranked by their likelihood.

In the KL-divergence retrieval model, both a query model $P_{\boldsymbol{q}}(\cdot)$ and a document model $P_{\boldsymbol{d}_h}(\cdot)$ are constructed. The documents in the collection are ranked according to the KL-divergence $KL(P_{\boldsymbol{q}}(\cdot)||P_{\boldsymbol{d}_h}(\cdot))$ between these models.

2.2 Language Model Construction

There are several ways to construct a query model and a document model. One method is maximum likelihood estimation (MLE). The MLE of a word w with respect to a text $\boldsymbol{t}$ (a query or document) is computed using

$$P_{\boldsymbol{t}}^{MLE}(w) = \frac{tf(w, \boldsymbol{t})}{|\boldsymbol{t}|}, \tag{1}$$

where $tf(w, \boldsymbol{t})$ represents the frequency of w in $\boldsymbol{t}$, and $|\boldsymbol{t}|$ represents the number of words in $\boldsymbol{t}$.

Dirichlet prior smoothing (DIR) (Zhai and Lafferty 2004) is a well-known construction method. The DIR of w with respect to $\boldsymbol{t}$ is computed as follows.

$$P_{\boldsymbol{t}}^{DIR}(w) = \frac{tf(w, \boldsymbol{t}) + \mu P_{\boldsymbol{D}_{all}}^{MLE}(w)}{|\boldsymbol{t}| + \mu}, \tag{2}$$

where $\boldsymbol{D}_{all}$ represents a collection, and μ represents the smoothing parameter that controls the degree of confidence for the frequency of w in $\boldsymbol{D}_{all}$ (rather than in the frequency in $\boldsymbol{t}$).

2.3 Fundamental RF Method

Zhai and Lafferty proposed a fundamental RF method for language modeling (Zhai and Lafferty 2001). When user feedback $\boldsymbol{F} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_G)$ is available, they construct a language model $P_{\boldsymbol{F}}(\cdot)$ for the feedback. Then, a new query model is constructed by interpolating the feedback model with the original query model that was used to obtain the initial search results. Finally, they modify the search results using the new query model.

Their experiments demonstrate that their proposed method is effective. However, they only use the information of words that appear in a text for the RF. In our proposed method, we also use information about words that do not appear in a text but are highly probable given the latent topics in the text.

3 LDA

In this section, we explain LDA, which the proposed method uses to determine words that are highly probable given the latent topics of a text.

3.1 Overview

LDA (Blei et al. 2003) is a popular topic model that is based on the idea that documents are generated from a mixture of topics, where a topic is a distribution of words. LDA posits that a topic proportion $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ for a document can take any value in the $(K-1)$ simplex. Therefore, a topic proportion is a point of the simplex as drawn from a Dirichlet distribution.

According to the generative process, the probability of document $\boldsymbol{d}$ in LDA is calculated as follows:

$$P(\boldsymbol{d}|\boldsymbol{\alpha}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K) = \int P(\boldsymbol{\theta}|\boldsymbol{\alpha}) \left(\prod_{j=1}^{J} \left(\sum_{k=1}^{K} P(w_j|z_k, \boldsymbol{\beta}_k)\, P(z_k|\boldsymbol{\theta}) \right)^{tf(w_j, \boldsymbol{d})} \right) d\boldsymbol{\theta}, \tag{3}$$

where $P(\boldsymbol{\theta}|\boldsymbol{\alpha})$ represents the probability of $\boldsymbol{\theta}$ drawn from a Dirichlet distribution, and the parameter $\boldsymbol{\alpha}$ is a K-vector with components $\alpha_k > 0$ $(k = 1, \ldots, K)$. z_k represents the k-th topic, and $\boldsymbol{\beta}_k$ represents a word distribution for the topic. $P(w_j|z_k, \boldsymbol{\beta}_k)$ represents the probability of w_j in z_k, and $P(z_k|\boldsymbol{\theta})$ represents the probability of z_k drawn from a multinomial distribution. J represents the number of words in a vocabulary.

3.2 Parameter Estimation

There are two ways to estimate the parameters: Gibbs sampling and a variational method (Griffiths and Steyvers 2004; Blei et al. 2003). Gibbs sampling is more popular because it determines better parameter values and is simpler to implement. However, Gibbs sampling for parameter estimation takes much longer to execute than variational methods. Therefore, we use a variational method because search systems should return results as quickly as possible.

First, variational parameters $\boldsymbol{\gamma}_i = (\gamma_{i1}, \dots, \gamma_{iK})$ and $\boldsymbol{\phi}_i = (\boldsymbol{\phi}_{i1}, \dots, \boldsymbol{\phi}_{iJ})$ are introduced for each document $\boldsymbol{d}_i$ $(i = 1, \dots, I)$ in the training data. Note that $\boldsymbol{\phi}_{ij} = (\phi_{ij1}, \dots, \phi_{ijK})$. Then, optimum parameter values are found by repeatedly computing the following pair of updated equations:

$$\phi_{ijk} \propto \beta_{kj} \exp\biggl(\Psi(\gamma_{ik}) - \Psi\Bigl(\sum_{k'=1}^{K} \gamma_{ik'}\Bigr)\biggr) \tag{4}$$

$$\gamma_{ik} = \alpha_k + \sum_{j=1}^{J} \phi_{ijk}\, tf(w_j, \boldsymbol{d}_i), \tag{5}$$

where Ψ is the first derivative of the log Γ function.

Next, α_k and $\boldsymbol{\beta}_k$ are updated using $\boldsymbol{\gamma}_i$ and $\boldsymbol{\phi}_i$. A Newton-Raphson method has been used to estimate each α_k (Blei et al. 2003). However, the fixed-point iteration method (Minka 2000) is a better estimation technique; therefore, we have used updated equations based on this method. The updated equations for α_k and $\boldsymbol{\beta}_k$, respectively, are as follows:

$$\beta_{kj} \propto \sum_{i=1}^{I} \phi_{ijk}\, tf(w_j, \boldsymbol{d}_i) \tag{6}$$

$$\alpha_k = \frac{\sum_{i=1}^{I} \{\Psi(\alpha_k + n_{ik}) - \Psi(\alpha_k)\}}{\sum_{i=1}^{I} \{\Psi(\alpha_0 + |\boldsymbol{d}_i|) - \Psi(\alpha_0)\}}\, \alpha_k^{old}, \tag{7}$$

where $n_{ik} = \sum_{j=1}^{J} \phi_{ijk}\, tf(w_j, \boldsymbol{d}_i)$, $\alpha_0 = \sum_{k'=1}^{K} \alpha_{k'}$, and α_k^{old} represents α_k before the update.

Finally, updates of $\boldsymbol{\gamma}_i$ and $\boldsymbol{\phi}_i$ for each $\boldsymbol{d}_i$ and those of α_k and $\boldsymbol{\beta}_k$ are iterated until convergence. When α_k and $\boldsymbol{\beta}_k$ have been estimated, we obtain the probability of document $\boldsymbol{d}_i$ using Eq. (3). In addition, we can obtain $P^{LDA}_{\boldsymbol{d}_i}(w_j)$ using the estimated $\boldsymbol{\gamma}_i$ and the following equation:

$$P^{LDA}_{\boldsymbol{d}_i}(w_j) \simeq \sum_{k=1}^{K} \frac{\beta_{kj}\gamma_{ik}}{\sum_{k'=1}^{K} \gamma_{ik'}}, \tag{8}$$

where $\gamma_{ik} / \sum_{k'=1}^{K} \gamma_{ik'}$ is a distribution over the latent topics of $\boldsymbol{d}_i$. We interpolate each $\boldsymbol{\beta}_{kj}$ according to the distribution and obtain the probabilities of w_j that are highly probable given $\boldsymbol{d}_i$.

3.3 Inference of Unseen Documents

LDA is considered a Bayesian extension of probabilistic latent semantic analysis (Hofmann 1999). A major advantage of LDA is that it can infer the probabilities of unseen documents that are not included in the training data. When we compute the probabilities of an unseen document $\boldsymbol{d}_{I+1}$, the variational parameters $\boldsymbol{\gamma}_{I+1}$ and $\boldsymbol{\phi}_{I+1}$ are estimated using Eqs. (4) and (5), respectively. The estimated values obtained using the training document set are used for α_k and $\boldsymbol{\beta}_k$. When $\boldsymbol{\gamma}_{I+1}$ has been estimated, we can obtain the probability $P^{LDA}_{\boldsymbol{d}_{I+1}}(w_j)$ using Eq. (8). In the proposed method, this advantage of LDA allows us to calculate the probabilities of words that are highly probable given the feedback.

3.4 LDA in IR

LDA has been successfully used in various fields such as natural language processing (Blei et al. 2003), image processing (Fei-Fei and Perona 2005), and speech recognition (Heidel, Chang, and Lee 2007). In the IR field, Wei and Croft (2006) incorporated LDA into a query likelihood model, and Yi and Allan (2009) have incorporated LDA into a document likelihood model. Zhou and Wade (2009) incorporated LDA into a KL-divergence retrieval model. These methods construct a language model for each document using LDA and the obtained search results for a given query according to their scores (e.g., query likelihood). In this study, we focus on user feedback. We construct a language model for the feedback using LDA, which we use to modify the search results.

4 Proposed Method

4.1 Overview

We propose an RF method that uses surface and latent information in texts. An overview of the proposed method is illustrated in Figure 1. We refer to the model containing surface and latent information in text $\boldsymbol{t}$ as a hybrid language model $P^{HYB}_{\boldsymbol{t}}(\cdot)$. First, a query $\boldsymbol{q}$ is submitted by a user and initial search results $\boldsymbol{D_q} = (\boldsymbol{d}_1, \ldots, \boldsymbol{d}_I)$ are obtained (**Step 1**). Next for each document $\boldsymbol{d}_i$ $(i = 1, \ldots, I)$ in $\boldsymbol{D_q}$, a hybrid language model $P^{HYB}_{\boldsymbol{d}_i}(\cdot)$ that contains the document's surface and latent information is constructed (**Step 2**). Then, when the user provides feedback $\boldsymbol{F} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_G)$, a hybrid language model $P^{HYB}_{\boldsymbol{F}}(\cdot)$ for $\boldsymbol{F}$ is constructed (**Step 3**). Finally, a new query model using the feedback model is constructed, and $\boldsymbol{D_q}$ is re-ranked using the new query model. Documents that have a hybrid language model resembling the user feedback are given a higher rank (**Step 4**). The details of each step are described in the following sections.

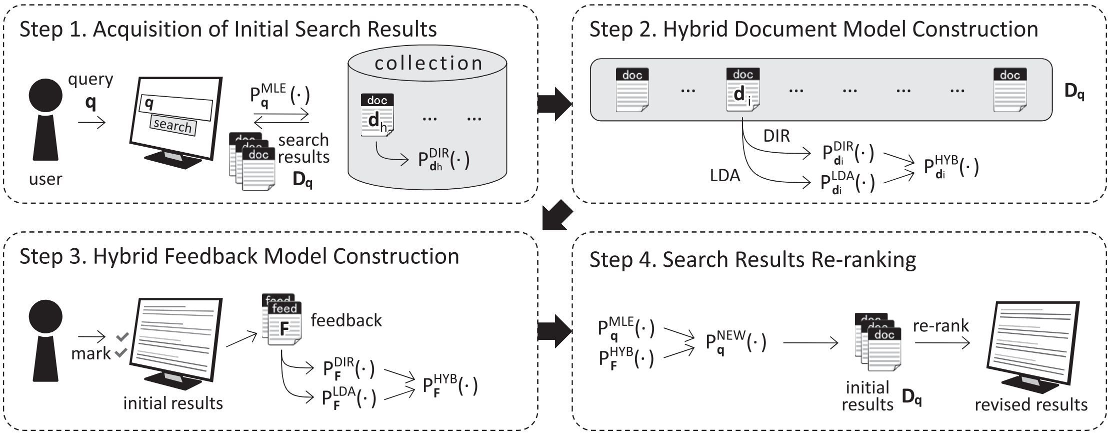

Fig. 1 Overview of the proposed method

It is possible to use latent semantic analysis (LSA) to take advantage of latent information in texts. We could use LSA on documents in a collection and map the documents to a lower dimensional semantic space. By doing this, the information of words that are highly probable given the documents can be used. However, this method requires significant time to execute because LSA must be applied to the entire collection and there may be tens of millions of documents. If documents are added or removed, LSA must be performed again on the entire collection. Another disadvantage of LSA is that it does not naturally infer the probabilities of unseen texts. In the proposed method, each time search results are obtained, LDA must be performed (Section 4.3). However, this does not take a long time because the number of documents in the search results is much smaller than that in the entire collection. In addition, LDA can infer the probabilities of unseen documents.

4.2 Acquisition of Initial Search Results

In the proposed method, we use a KL-divergence retrieval model (Lafferty and Zhai 2001) to obtain the initial search results for a given query. For each document $\boldsymbol{d}_h$ $(h = 1, \ldots, H)$ in the collection $\boldsymbol{D}_{all}$, a DIR-based document model $P_{\boldsymbol{d}_h}^{DIR}(\cdot)$ is constructed in advance. For a given query $\boldsymbol{q}$, we construct the MLE-based query model $P_{\boldsymbol{q}}^{MLE}(\cdot)$. Then, for each document containing $\boldsymbol{q}$ in $\boldsymbol{D}_{all}$, we compute the KL-divergence between the DIR-based document model and the MLE-based query model. That is, we define the score of a document $\boldsymbol{d}_h$ for a query $\boldsymbol{q}$ as

$$initial\_score(\boldsymbol{d}_h, \boldsymbol{q}) = -KL(P_{\boldsymbol{q}}^{MLE}(\cdot)||P_{\boldsymbol{d}_h}^{DIR}(\cdot)). \tag{9}$$

The initial search results $\boldsymbol{D}_q$ are obtained by ranking the documents according to their scores.

We use MLE to construct the query model (e.g., Zhai and Lafferty 2001). When a query model is constructed using MLE, the ranking based on the KL-divergence retrieval model is equivalent to that based on the query likelihood model (Ponte and Croft 1998).

4.3 Hybrid Document Model Construction

We construct a hybrid language model $P_{\boldsymbol{d}_i}^{HYB}(\cdot)$ for each document $\boldsymbol{d}_i\,(i = 1, \ldots, I)$ in $\boldsymbol{D_q}$. In this model, we consider both the surface and latent information in the documents.

First, an LDA-based document model $P_{\boldsymbol{d}_i}^{LDA}(\cdot)$ is constructed for each $\boldsymbol{d}_i$. We perform LDA on $\boldsymbol{D_q}$ to infer the topic distribution in each $\boldsymbol{d}_i$, and estimate the parameters α_k, $\boldsymbol{\beta}_k$ $(k = 1, \ldots, K)$ and $\boldsymbol{\gamma}_i$ for each $\boldsymbol{d}_i$. Then, we construct $P_{\boldsymbol{d}_i}^{LDA}(\cdot)$ using the estimated parameters and Eq. (8). As described in Section 3.2, $P_{\boldsymbol{d}_i}^{LDA}(\cdot)$ is constructed on the basis of highly probable words given the latent topics in $\boldsymbol{d}_i$.

Next, for each $\boldsymbol{d}_i$, we construct $P_{\boldsymbol{d}_i}^{HYB}(\cdot)$ by interpolating the constructed $P_{\boldsymbol{d}_i}^{LDA}(\cdot)$ with $P_{\boldsymbol{d}_i}^{DIR}(\cdot)$ as follows.

$$P_{\boldsymbol{d_i}}^{HYB}(w) = (1-a)P_{\boldsymbol{d_i}}^{DIR}(w) + aP_{\boldsymbol{d_i}}^{LDA}(w), \tag{10}$$

where $0 \leq a \leq 1$. $P_{\boldsymbol{d}_i}^{DIR}(\cdot)$ is constructed using the words that appear in $\boldsymbol{d}_i$. By interpolating the two models, our method constructs a document model that contains the surface and latent information in $\boldsymbol{d}_i$.

4.4 Hybrid Feedback Model Construction

We construct a hybrid language model $P_{\boldsymbol{F}}^{HYB}(\cdot)$ for user feedback. First, we perform LDA on $\boldsymbol{F}$ to infer the topic distribution in $\boldsymbol{F}$ and estimate the variational parameters for $\boldsymbol{F}$, as described in Section 3.3. Then, we construct the LDA-based feedback model $P_{\boldsymbol{F}}^{LDA}(\cdot)$ using the estimated parameters and Eq. (8). Similar to $P_{\boldsymbol{d}_i}^{LDA}(\cdot)$, $P_{\boldsymbol{F}}^{LDA}(\cdot)$ is constructed on the basis of the highly probable words from the latent topics in $\boldsymbol{F}$. Finally, $P_{\boldsymbol{F}}^{HYB}(\cdot)$ is constructed in the same manner as $P_{\boldsymbol{d}_i}^{HYB}(\cdot)$.

$$P_{\boldsymbol{F}}^{HYB}(w) = (1-a)P_{\boldsymbol{F}}^{DIR}(w) + aP_{\boldsymbol{F}}^{LDA}(w), \tag{11}$$

where $P_{\boldsymbol{F}}^{DIR}(\cdot)$ is constructed using Eq. (2). By interpolating the two models, our method constructs a feedback model that contains the surface and latent information in $\boldsymbol{F}$.

4.5 Re-ranking Search Results

We construct a new query model that is used to re-rank the initial search results $\boldsymbol{D_q}$. The new query model $P_{\boldsymbol{q}}^{NEW}(\cdot)$ is constructed by interpolating the original query model $P_{\boldsymbol{q}}^{MLE}(\cdot)$ with the hybrid feedback model $P_{\boldsymbol{F}}^{HYB}(\cdot)$ as follows:

$$P_{\boldsymbol{q}}^{NEW}(w) = (1-b)P_{\boldsymbol{q}}^{MLE}(w) + bP_{\boldsymbol{F}}^{HYB}(w), \tag{12}$$

where $0 \leq b \leq 1$.

Then, for each $\boldsymbol{d}_i$ in $\boldsymbol{D_q}$, we compute the KL-divergence between $P_{\boldsymbol{d}_i}^{HYB}(\cdot)$ and $P_{\boldsymbol{q}}^{NEW}(\cdot)$. That is, the score of document $\boldsymbol{d}_i$ for query $\boldsymbol{q}$ (given feedback $\boldsymbol{F}$) is defined as

$$\textit{re-ranking\_score}(\boldsymbol{d}_i, \boldsymbol{q}, \boldsymbol{F}) = -KL(P_{\boldsymbol{q}}^{NEW}(\cdot)||P_{\boldsymbol{d}_i}^{HYB}(\cdot)). \tag{13}$$

Finally, we obtain the revised search results by re-ranking the documents in $\boldsymbol{D_q}$ according to their scores.

5 Experiments

In this section, we present the results of experiments performed to evaluate the proposed method.

5.1 Test Set

In the experiments, we used the test set from the Web Retrieval Task from the Third NTCIR Workshop (Eguchi, Oyama, Ishida, Kuriyama, and Kando 2002). The test set consists of 11,038,720 Japanese web pages and 47 information needs. For each information need, approximately 2,000 documents are rated as highly relevant, fairly relevant, partially relevant or irrelevant. We can evaluate the ranking of search results using these annotated documents.

Figure 2 gives an example of an information need for the Web Retrieval Task. The meaning of each element is given below.

⟨NUM⟩ 0042 ⟨/NUM⟩

⟨TITLE⟩ Easter, Christ ⟨/TITLE⟩

⟨DESC⟩ Find documents describing Easter, a celebration of the resurrection of Christ ⟨/DESC⟩

⟨RDOC⟩ NW002542912, NW002008347, NW000837198 ⟨/RDOC⟩

Fig. 2 Example information need from the Third NTCIR Workshop.

NUM Identification number of the information need.

TITLE Query submitted to a search engine. Up to three words are provided, listed according to importance.

DESC A description of the user's information need in a single sentence.

RDOC Up to three identification numbers referring to examples of documents that are relevant to this information need, listed according to importance.

In our experiments, we used the first two terms in the ⟨TITLE⟩ tag as the query. We collected each document containing a query (Section 4.2). When we used all the terms in the ⟨TITLE⟩, there were too few documents in the search results. For example, for identification numbers 0027, 0047, and 0058, we could only obtain 17, 5, and 14 documents, respectively. For identification number 0061, we could not find any documents. This caused unreliable evaluation results. In other words, if an RF method improved the ranking of the initial search results, the improvement was not reflected by evaluation measures such as precision at 10 (P@10), and we could not determine how well the method performed. Thus, to obtain a sufficient number of documents, we used only the first two terms in the ⟨TITLE⟩ tag. It should be noted that greater than 100 documents was defined as sufficient.

We used the documents in the ⟨RDOC⟩ tag as the user feedback. These are relevant documents selected by assessors and can be treated as user feedback. They were not necessarily included in the initial search results. In such cases, one may think that these documents should not be used as user feedback. However, when we evaluated the RF method, we removed these documents from the search results regardless of whether they were included in the initial results (Section 5.3). In other words, whether these documents were included in the search results was not important because we removed them from the initial search results and the re-ranked results.

We did not use seven of the information needs (identification numbers: 0011, 0018, 0032, 0040, 0044, 0047, and 0061) because we were unable to retrieve a sufficient number of documents (i.e., 100 documents) for them even when using the first two terms as the query. We divided the remaining 40 information needs into development and test data. The development data consisted of 8 information needs (identification numbers 0008–0017), which were used to tune our method. The test data consisted of 32 information needs (identification numbers 0018–0063), which were used to evaluate our method.

5.2 Search Engine

In these experiments, we implemented a search engine that obtained the initial search results for a given query and re-ranked them using the proposed method. The details of the implementation are as follows.

We used the $11,038,720$ web pages in our test set as the collection (i.e., $\boldsymbol{D}_{all}$). Each document was converted into the format presented by Shinzato, Kawahara, Hashimoto, and Kurohashi (2008). In this format, each sentence in a document was segmented into words. Each word was given a representative form using JUMAN (Kurohashi, Nakamura, Matsumoto, and Nagao 1994), a Japanese morphological analyzer. Then, we constructed the DIR-based document model for each document. We set the smoothing parameter $\mu = 1,000$, which is consistent with previous studies (Zhai and Lafferty 2001; Wei and Croft 2006; Yi and Allan 2009).

When given a query, we converted its terms into a representative form using JUMAN, constructed its MLE-based language model, and obtained the initial search results by ranking documents in the collection.

The LDA configuration is given below. We set the initial values of $\alpha_k\,(k = 1, \ldots, K)$ to 1, and the initial value of each $\beta_{kj}\,(k = 1, \ldots, K,\ j = 1, \ldots, J)$ to random values. The number of iterations for the variational parameters and that for α_k and $\boldsymbol{\beta}_k$ were set to 10. We limited the size of the vocabulary in LDA, denoted as J, to 100. We selected 100 words on the basis of their importance in the search results. Note that the importance of a word w to the search results $\boldsymbol{D_q}$ is defined as $df(w, \boldsymbol{D_q}) * \log(|H|/df(w, \boldsymbol{D}_{all}))$, where $df(w, \boldsymbol{D})$ represents the document frequency of w in documents $\boldsymbol{D}$.

5.3 Evaluation Method

We removed the feedback documents from both the initial and re-ranked results. We can evaluate the performance of an RF method by comparing the initial search results with the re-ranked results. A common evaluation problem is how to handle documents that users have marked as relevant (Hull 1993). If the initial and re-ranked results are compared in a straightforward manner, the latter have an advantage because documents that are known to be relevant tend to be re-ranked higher. However, if we remove them from the re-ranked results, they are disadvantaged. This is especially true if there are few relevant documents. Therefore, we removed the documents used as user feedback from both results. This allowed us to make a fair comparison between the initial and re-ranked results.

We evaluated the method using P@10, mean average precision (MAP), and normalized discounted cumulative gain for the 10 top (re-)ranked documents (NDCG@10) (Järvelin and Kekäläinen 2002). When calculating P@10 and MAP, documents that were rated as highly relevant, fairly relevant, and partially relevant were regarded as relevant, while documents rated as irrelevant and unrated documents were regarded as irrelevant. When calculating NDCG, we assessed the relevance score of documents rated highly relevant, fairly relevant, and partially relevant as 3, 2, and 1, respectively.

5.4 Performance of the Proposed Method

We examined the effectiveness of the proposed method in re-ranking the initial search results using explicit feedback. As described in Section 5.1, we used the test data and the first two terms in the ⟨TITLE⟩ as the query for each information need. We defined the initial search results as the 100 documents with the highest initial scores and then re-ranked them using the proposed method. We used the first two documents in the ⟨RDOC⟩ tag for each information need as the explicit feedback. The average number of words in a document was $3,589$. We set $(a, b, K) = (0.2, 0.9, 50)$ because this setting obtained the best results in the preliminary experiment described in Section 5.6.

The results are listed in Table 1(a). INIT represents the ranking of the initial search results, and OURS represents the re-ranked results using the proposed method. For comparison, we also show the performance of some baseline methods, ZHAI represents the method by Zhai and Lafferty (2001) and OURS ($a = 0.0$) represents the proposed method without latent information. ZHAI is essentially the same as OURS ($a = 0.0$). Both methods construct the feedback model by modifying the surface word distribution of the feedback using that of the collection. The difference lies in the way the word distribution is modified. The former uses an expectation-maximization algorithm for the modification, while the latter uses DIR estimation. In OURS ($a = 0.0$), we set $b = 0.5$. This value was determined in the preliminary experiment.

DIC also represents a baseline method. The proposed method uses information about words that are highly probable given a text. Dictionaries of synonyms and related words can also be used for this purpose. DIC is an extension of OURS ($a = 0.0$); DIC uses synonyms of the

Table 1 Comparative performance of the proposed method

(a) Effectiveness with respect to explicit RF

	P@10	MAP	NDCG@10
INIT	0.275	0.110	0.209
ZHAI	0.297	0.114	0.218
OURS ($a = 0.0$)	0.294	0.115	0.221
DIC	0.300	0.114	0.225
OURS	0.351	0.148	0.271

(b) Effectiveness with respect to pseudo RF

	P@10	MAP	NDCG@10
INIT	0.294	0.116	0.233
ZHAI	0.294	0.117	0.242
OURS ($a = 0.0$)	0.291	0.117	0.239
DIC	0.300	0.117	0.245
OURS	0.318	0.115	0.260

surface words from the feedback and search results. For this method, we constructed a synonym dictionary from Japanese dictionaries using the method proposed by Shibata, Odani, Harashima, Oonishi, and Kurohashi (2008). For the Japanese dictionaries, we used Reikai Shougaku Kokugo Jiten (Tajika 2001) and Iwanami Kokugo Jiten (Nishio, Iwabuchi, and Mizutani 2002). The constructed dictionary contained 4,597 entries (e.g., “computer” = “electronic brain”).

The results shown in Table 1(a) indicate that OURS outperformed INIT for all metrics. For example, the proposed method improved the initial search results by 27.6% in P@10. These results suggest that the proposed method effectively re-ranked the initial search results. In addition, the proposed method outperformed ZHAI and OURS ($a = 0.0$), which do not use latent information from texts. This suggests that latent information is useful when re-ranking search results.

We investigated further and confirmed that the proposed method made good use of the words that did not appear in the feedback but were considered highly probable. Consider the information need shown in Figure 2. The user feedback did not contain the words “religion,” “holiday,” or “bible,” which are related to the information need. As such, ZHAI and OURS ($a = 0.0$) could not use these words. In contrast, these highly likely words had a certain degree of probability in the hybrid language model even though the words did not appear in the feedback. For example, the method could allocate the probabilities 0.0046, 0.0037, and 0.0024 to the words “religion,” “holiday,” and “bible,” respectively. The probabilities allocated to the words “Christmas” and “easter,” which appeared once in the feedback, were 0.0093 and 0.0060, respectively. Using these probabilities, the proposed method raised the score of documents that contained these words.

Although DIC outperformed ZHAI and OURS ($a = 0.0$), it did not outperform OURS. This may be due to the coverage of the synonym dictionary. DIC may perform better if wide-coverage synonym dictionaries are used. However, constructing such dictionaries is difficult. The proposed method also needs to know that a word is related to other words; however, unlike DIC, it does not need any dictionaries. Using LDA, the proposed method dynamically acquires the required knowledge from the search results. Consider the query “Mac price” described in Section 1. Suppose that the search results contain words such as “CPU,” “HDD,” “hamburger,” and “potato.” Our method performs LDA on the search results and dynamically acquires the knowledge that “CPU” is related to “HDD,” and “hamburger” is related to “potato.” In addition, even if “HDD” does not appear in a document, the proposed method can assign a certain degree of probability to the word from other related words, such as “CPU.” Thus, the proposed method does not require any dictionaries.

The proposed method can also be applied to pseudo RF. In pseudo RF, the top n documents in the initial search results are assumed to be relevant, and the search results are re-ranked on the basis of this assumption. We implemented pseudo RF using the proposed method for $n = 10$ and compared the initial results with the re-ranked results. Note that there are no relevant documents in pseudo RF. Thus, we evaluated the performance of each method using the raw (re-)ranked results.

The evaluation results are shown in Table 1(b). The results for INIT in this table differ from those in Table 1(a) because the documents used as feedback were not removed from the search results. The proposed method improved the initial search results in P@10 and NDCG@10. For example, the proposed method improved the initial search results by 8.2% in P@10. These results demonstrate that the proposed method is a promising candidate for pseudo RF as well as explicit RF.

5.5 Effect of the Amount of Feedback

In the second experiment, we simulated a situation where only a small amount of user feedback can be obtained. We investigated how the amount of feedback affected the performance of the proposed method. In practice, large quantities of user feedback are rarely available. Thus, an RF method should perform well only when a small amount of user feedback is available. We incrementally reduced the amount of explicit feedback and observed the change in P@10.

For this experiment, we used seven different quantities of explicit feedback: $G = 2^1$, 2^0, 2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, and 2^{-5} relevant documents. Note that, for example, $G = 2^1$ means that we used two relevant documents as user feedback. $G = 2^{-1}$ represents the use of half a document, i.e., half the words from the feedback were randomly sampled and only these words were used for RF. This allowed us to consider the case where part of a document (e.g., title or snippet) is given as user feedback.

Figure 3 shows the effect of the amount of feedback on the performance of the proposed method. For comparison, we also present the results of the proposed method without the latent information (OURS ($a = 0.0$)). INIT represents the precision of the ranking of the initial search results. From Figure 3, it is evident that the proposed method performed consistently well. For example, when one relevant document was given as user feedback, the proposed method improved the initial search results by 24.5% in P@10. In addition, the proposed method achieved a 5.3% improvement with only 2^{-5} documents. When $G = 2^{-5}$, there were on average 57 words in the feedback, i.e., $|\boldsymbol{F}|$ was 57. On the other hand, when G was small, the improvements achieved by OURS ($a = 0.0$) were negligible. As G becomes smaller, the amount of available surface information becomes smaller and the method can not improve the initial search results. On the other hand, OURS uses both surface and latent information. Even when only a small amount

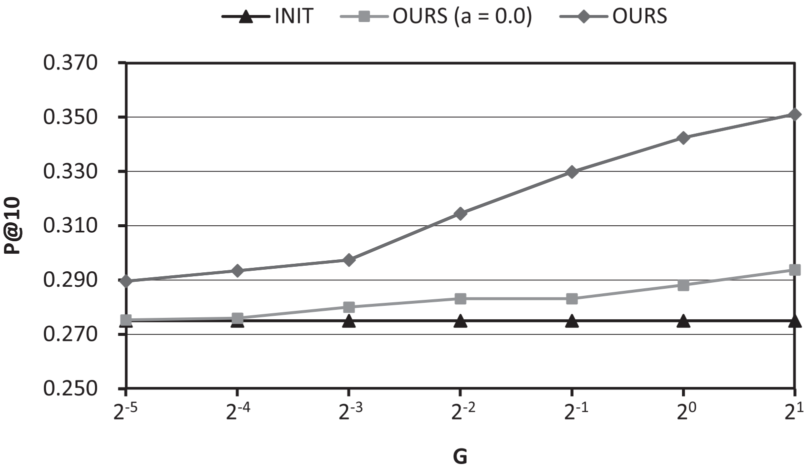

Fig. 3 Effect of the amount of feedback

of feedback is available, OURS improved the initial search results because the method uses more information.

5.6 Sensitivity to Parameters

The proposed method uses the following three parameters: a, b, and K. Parameter a controls the reliability of an LDA-based language model in a hybrid language model. Parameter b controls the reliability of a feedback model in a new query model, and parameter K represents the number of topics. In our experiments, we set $(a, b, K) = (0.2, 0.9, 50)$ for OURS and $b = 0.5$ for OURS $(a = 0.0)$. These values were determined in a preliminary experiment.

In the preliminary experiment, we re-ranked the initial search results using different parameter values and measured the changes in performance using the proposed method. We used the development data, applied our method with a and b ranging from 0.0 to 1.0 in steps of 0.1 and K ranging from 10 to 100 in steps of 10, and obtained the average of P@10 for all information needs. We used the first two terms in the ⟨TITLE⟩ as the query and the first two documents in the ⟨RDOC⟩ as the user feedback.

The results of the preliminary experiment are shown in Table 2 and Figure 4. Table 2 summarizes the results with respect to (a, b). The value of each cell in the table is the average of P@10 obtained for each K. For example, the value of cell $(a, b) = (0.1, 0.2)$ denotes that the average of P@10 obtained for $(a, b, K) = (0.1, 0.2, 10), (0.1, 0.2, 20), \ldots,$ $(0.1, 0.2, 100)$ is 0.286. The highest value in each column is shown in bold, and the highest value in each row is underlined.

As can be seen from the table, the proposed method achieved the best performance with

Table 2 Sensitivity to (a, b)

		b										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
a	0.0	**0.288**	**0.300**	**0.313**	**0.338**	**0.325**	0.338	0.325	0.313	0.325	0.325	0.063
	0.1	0.261	0.275	0.286	0.301	0.316	**0.340**	**0.369**	0.400	0.430	**0.462**	**0.455**
	0.2	0.235	0.254	0.274	0.293	0.308	0.332	0.368	**0.408**	**0.452**	**0.462**	0.433
	0.3	0.211	0.227	0.253	0.277	0.299	0.326	0.366	0.404	**0.452**	0.456	0.435
	0.4	0.195	0.212	0.231	0.265	0.292	0.321	0.360	0.403	0.437	0.455	0.437
	0.5	0.185	0.202	0.221	0.251	0.286	0.316	0.356	0.398	0.437	0.456	0.439
	0.6	0.174	0.192	0.214	0.240	0.283	0.310	0.356	0.394	0.432	0.452	0.444
	0.7	0.167	0.182	0.203	0.233	0.279	0.306	0.355	0.393	0.430	0.444	0.449
	0.8	0.156	0.173	0.193	0.224	0.270	0.304	0.345	0.389	0.425	0.440	0.446
	0.9	0.152	0.168	0.184	0.219	0.262	0.302	0.343	0.384	0.421	0.436	0.440
	1.0	0.148	0.163	0.179	0.215	0.253	0.295	0.345	0.389	0.413	0.427	0.431

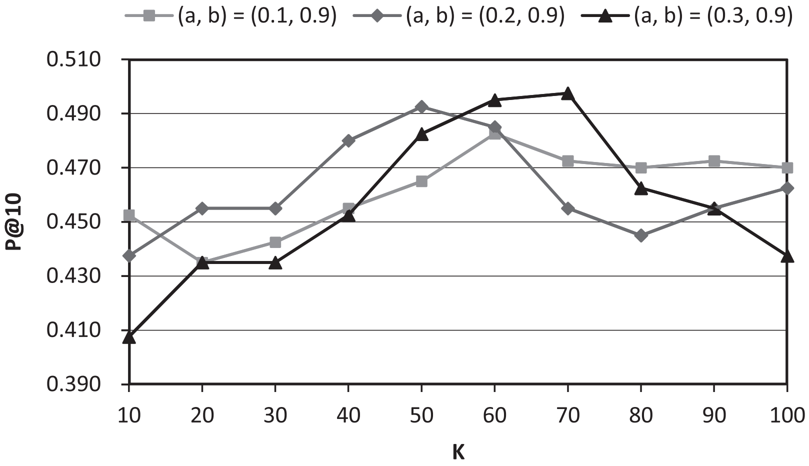

Fig. 4 Sensitivity to K

$(a, b) = (0.1, 0.9)$ or $(0.2, 0.9)$. The results with $a = 0.0$ (not using latent information) demonstrate that the method performed well with $b = 0.3$–0.5. On the other hand, the results with $a \geq 0.1$ (using latent information) demonstrate that the proposed method performed well with $b = 0.8$–1.0. The value of b (and the performance with this b) with $a \geq 0.1$ is greater than that with $a = 0.0$. This result suggests that latent information increases the reliability of the feedback model.

Figure 4 illustrates the effect of K on the performance of the proposed method. For clarity, we only show the results with $(a, b) = (0.1, 0.9), (0.2, 0.9)$, and $(0.3, 0.9)$, which produced good results (Table 2). According to the result presented in Figure 4, the proposed method performed well with $K = 50$–70. These results indicate that the proposed method achieves the best performance with $(a, b) = (0.1, 0.9)$ or $(0.2, 0.9)$, and $K = 50$–70.

5.7 Execution Time

In the proposed method, we perform LDA on the search results to construct an LDA-based document model for each document. We also perform LDA on the feedback to construct an LDA-based feedback model. Here, we report the execution times of these procedures.

In our experiments, the proposed method took 13.1–16.0 secs to perform LDA on the search results (100 documents). Note that we implemented LDA using Perl and C. We believe that this is an acceptable execution time because users typically browse documents in search results to select relevant documents. This process generally takes at least 1 min. In other words, we can perform LDA on the search results while the users are browsing the documents. Thus, LDA can be completed before the users re-rank the initial search results.

However, the number of retrieved documents is sometimes greater than 100. If a large number of documents is retrieved, LDA requires a significant amount of time to execute. One way to avoid this problem is to use only the top ranked documents as the search results. The time required for LDA is not a matter of concern if we use the top 100 documents. Another alternative is to parallelize the estimation of the variational parameters in LDA, a process that takes the majority of the execution time. Variational parameters for each document are independent of those for other documents. Thus, we can parallelize the estimation of the parameters and reduce the time required for the procedure. For example, Nallapati, Cohen, and Lafferty (2007) achieved a 14.5 times speedup using 50 cluster nodes. Therefore, it is expected that we can reduce the execution time based on these studies.

It took less than 1 sec to perform LDA on the feedback. For example, the execution time was only 0.1–0.2 secs when using two documents as feedback. Thus, the time required to perform LDA on the feedback is negligible.

6 Conclusion

We have proposed an RF method using surface and latent information from texts and investigated its effectiveness. Using LDA, our method infers the distributions over words that are highly probable given the user feedback and each document in the search results. Then, a hybrid word distribution is constructed by interpolating the latent word distribution with the surface word distribution for the feedback and each document. Finally, documents whose hybrid word distributions resemble the feedback are regarded as relevant to the user's information need, and are re-ranked higher. Through our experiments, we confirmed that the proposed method performs well for both explicit and pseudo RF. The proposed method also performs well when only a small amount of feedback is available.

In future, we intend to use negative feedback in the proposed method. In this study, only use positive feedback (relevant documents for a query) was used for re-ranking search results; however, we believe that negative feedback (irrelevant documents) can also be useful.

Reference

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). "Latent Dirichlet Allocation." *Jounal of Machine Learning Research*, **3**, pp. 993–1022.

Eguchi, K., Oyama, K., Ishida, E., Kuriyama, K., and Kando, N. (2002). "The Web Retrieval Task and its Evaluation in the Third NTCIR Workshop." In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pp. 375–376.

Fei-Fei, L. and Perona, P. (2005). "A Bayesian Hierarchical Model for Learning Natural Scene Categories." In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pp. 524–531.

Griffiths, T. L. and Steyvers, M. (2004). "Finding scientific topics." In *Proceedings of the National Academy of Sciences of the United States of America (NAS)*, pp. 5228–5235.

Heidel, A., Chang, H., and Lee, L. (2007). "Language Model Adaptation using Latent Dirichlet Allocation and an Efficient Topic Inference Algorithm." In *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH 2007)*, pp. 2361–2364.

Hofmann, T. (1999). "Probabilistic Latent Semantic Analysis." In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI 1999)*, pp. 289–296.

Hull, D. (1993). "Using Statistical Testing in the Evaluation of Retrieval Experiments." In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1993)*, pp. 329–338.

Ide, E. (1971). "New Experiments in Relevance Feedback." In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 337–354. Prentice-Hall Inc.

Jansen, B. J., Spink, A., and Saracevic, T. (2000). "Real Life, Real Users, and Real Needs: A

Study and Analysis of User Queries on the Web." *Information Processing and Management*, **36** (2), pp. 207–227.

Järvelin, K. and Kekäläinen, J. (2002). "Cumulated Gain-Based Evaluation of IR Techniques." *ACM Transactions on Information Systems*, **20** (4), pp. 422–446.

Kurohashi, S., Nakamura, T., Matsumoto, Y., and Nagao, M. (1994). "Improvements of Japanese Morphological Analyzer JUMAN." In *Proceedings of the International Workshop on Sharable Natural Language Resources (SNLR)*, pp. 22–28.

Lafferty, J. and Zhai, C. (2001). "Document Language Models, Query Models, and Risk Minimization for Information Retrieval." In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pp. 111–119.

Lavrenko, V. and Croft, W. B. (2001). "Relevance-Based Language Models." In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pp. 120–127.

Minka, T. P. (2000). "Estimating a Dirichlet distribution." Tech. rep., Microsoft.

Nallapati, R., Cohen, W., and Lafferty, J. (2007). "Parallelized Variational EM for Latent Dirichlet Allocation: An Experimental Evaluation of Speed and Scalability." In *Proceedings of the 7th IEEE International Conference on Data Maining Workshops (ICDMW 2007)*, pp. 349–354.

Nishio, M., Iwabuchi, E., and Mizutani, S. (2002). *Iwanami Kokugo Jiten.* Iwanami Shoten.

Ponte, J. M. and Croft, W. B. (1998). "A Language Modeling Approach to Information Retrieval." In *Proceedings of the 21nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pp. 275–281.

Rocchio, J. J. (1971). "Relevance Feedback in Information Retrieval." In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 313–323. Prentice-Hall Inc.

Salton, G., Wong, A., and Yang, C.-S. (1975). "A Vector Space Model for Automatic Indexing." *Communications of the ACM*, **18** (11), pp. 613–620.

Shibata, T., Odani, M., Harashima, J., Oonishi, T., and Kurohashi, S. (2008). "SYNGRAPH: A Flexible Matching Method Based on Synonymous Expression Extraction from an Ordinary Dictionary and a Web Corpus." In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pp. 787–792.

Shinzato, K., Kawahara, D., Hashimoto, C., and Kurohashi, S. (2008). "A Large-Scale Web Data Collection as a Natural Language Processing Infrastructure." In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pp. 2236–2241.

Spärck Jones, K., Walker, S., and Robertson, S. E. (2000). "A Probabilistic Model of Information Retrieval: Development and Comparative Experiments." *Information Processing and Management*, **36** (6), pp. 779–808, 809–840.

Tajika, J. (2001). *Sanseido Reikai Shougaku Kokugo Jiten*. Sanseido.

Wei, X. and Croft, W. (2006). "LDA-Based Document Models for Ad-hoc Retrieval." In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pp. 178–185.

Yi, X. and Allan, J. (2009). "A Comparative Study of Utilizing Topic Models for Information Retrieval." In *Proceedings of the 31st European Conference on Information Retrieval (ECIR 2009)*, pp. 29–41.

Zhai, C. and Lafferty, J. (2001). "Model-based Feedback in the Language Modeling Approach to Information Retrieval." In *Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM 2001)*, pp. 403–410.

Zhai, C. and Lafferty, J. (2004). "A Study of Smoothing Methods for Language Models Applied to Information Retrieval." *ACM Transactions on Information Systems*, **22** (2), pp. 179–214.

Zhou, D. and Wade, V. (2009). "Latent Document Re-Ranking." In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pp. 1571–1580.

Jun Harashima: received his B.Eng. in 2007, and M.Sc. and Ph.D. in Informatics in 2009 and 2013, from Kyoto University. He is currently employed as an engineer at COOKPAD Inc. His research interests include natural language processing and information retrieval.

Sadao Kurohashi: received his Ph.D. in Electrical Engineering from Kyoto University in 1994. He is currently a professor of the Graduate School of Informatics at Kyoto University. His research interests include natural language processing, knowledge acquisition/representation, and information retrieval.

(Received December 2, 2011)
(Revised March 6, 2012)
(Accepted May 14, 2012)