テキストの表層情報と潜在情報を利用した適合性フィードバック

原島　純[†]・黒橋　禎夫[†]

適合性フィードバックの手法の多くは，テキストに表層的に出現する単語の情報だけを用いて検索結果をリランキングしている．これに対し，本稿では，テキストに表層的に出現する単語の情報だけでなく，テキストに潜在的に現れうる単語の情報も利用する適合性フィードバックの手法を提案する．提案手法では，まず検索結果に対して Latent Dirichlet Allocation (LDA) を実行し，各文書に潜在する単語の分布を推定する．ユーザからフィードバックが得られたら，これに対しても LDA を実行し，フィードバックに潜在する単語の分布を推定する．そして，表層的な単語の分布と潜在的な単語の分布の両方を用いてフィードバックと検索結果中の各文書との類似度を算出し，これに基づいて検索結果をリランキングする．実験の結果，2 文書（合計 3,589 単語）から成るフィードバックが与えられたとき，提案手法が初期検索結果の Precision at 10 (P@10) を 27.6% 改善することが示された．また，提案手法が，フィードバックが少ない状況でも，初期検索結果のランキング精度を改善する特性を持つことが示された（e.g., フィードバックに 57 単語しか含まれていなくても，P@10 で 5.3% の改善が見られた）．

キーワード：情報検索，適合性フィードバック，LDA

Relevance Feedback using Surface and Latent Information in Texts

Jun Harashima[†] and Sadao Kurohashi[†]

Most of the previous relevance feedback methods re-rank search results using only the information of surface words in texts. In this paper, we present a novel method that uses not only the information of surface words, but also that of latent words that are highly probable from the texts. In the proposed method, we infer the latent word distribution in each document in the search results using latent Dirichlet allocation (LDA). When user 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 then, we re-rank the search results based on the similarities. Evaluation results show that when user feedback that consists of two documents (3,589 words) is given, our method improves the initial search results by 27.6% in precision at 10 (P@10). Additionally, it proves that our method has the advantage of performing well even when only a small amount of user feedback is available (e.g., improvement of 5.3% in P@10 was achieved even when user feedback constituted only 57 words).

[†] 京都大学大学院情報学研究科, Graduate School of Informatics, Kyoto University

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

1 はじめに

検索エンジンの主な目的は，ユーザの情報要求に適合する文書をランキング形式でユーザに提供することである．しかし，情報要求に見合うランキングを実現するのは容易ではない．これは，ユーザが入力するクエリが一般的に，短く，曖昧であり (Jansen, Spink, and Saracevic 2000)，ユーザの情報要求を推定するのが困難であることに起因する．例えば「マック ␣ 価格」というクエリは，「Mac（コンピュータ）」の価格とも，「マクドナルド」の価格とも，もしくは他の「マック」の価格とも解釈できる．そのため，どの「マック」に関する文書が求められているのか分からなければ，ユーザの情報要求に見合うランキングを実現するのは難しい．

このような問題を解決する方法の一つとして，適合性フィードバック (Rocchio 1971) がある．適合性フィードバックでは，ユーザから明示的（もしくは擬似的）に得られるフィードバックを利用することで，検索結果のランキングを修正する．具体的には，次のような手続きに従ってランキングの修正を行う．

(1) クエリに対する初期検索結果をユーザに提示する．
(2) 初期検索結果中から，情報要求に適合する文書をユーザに選択させる．
(3) 選択された文書（フィードバック）を利用して，初期検索結果のランキングを修正する．

例えば，「Mac（コンピュータ）」の価格に関する文書がフィードバックとして得られれば，ユーザがこの話題に関心を持っていると推測できる．そして，この情報を基に検索結果のランキングを修正することができる．

適合性フィードバックには，ベースとするランキングアルゴリズムに応じて，様々な手法がある．Rocchio の手法 (Rocchio 1971) や Ide の手法 (Ide 1971) は，ベクトル空間モデルに基づくランキングアルゴリズム (Salton, Wong, and Yang 1975) に対する適合性フィードバックの手法として有名である．確率モデルに基づくランキングアルゴリズム (Spärck Jones, Walker, and Robertson 2000) においては，フィードバックを用いて，クエリ中の単語の重みを修正したり，クエリを拡張することができる．言語モデルに基づくランキングアルゴリズム (Ponte and Croft 1998) に対しては，Zhai らの手法 (Zhai and Lafferty 2001) が代表的である．

このように適合性フィードバックには様々な手法があるが，それらの根底にあるアイディアは同じである．すなわち，適合性フィードバックでは，フィードバックと類似する文書を検索結果の上位にリランキングする．ここで，既存の手法の多くは，テキスト（フィードバック及び検索結果中の各文書）に表層的に出現する単語の情報だけを用いて類似度を算出している．すなわち，テキストに含まれていない単語の情報は利用していない．しかし，表層的には出現していなくても，そのテキストに潜在的に現れうる単語の情報は，リランキングに役に立ちうる

と考えられる．上の「マック」の例であれば，仮にフィードバック（この例では「Mac（コンピュータ）」の価格に関する文書）に「CPU」や「ハードディスク」などの単語が含まれていなくても，これらの単語はフィードバックとよく関連しており，潜在的にはフィードバックに現れうる．検索結果中の適合文書（i.e.,「Mac（コンピュータ）」の価格に関する文書）についても同様のことが言える．仮にある適合文書にこれらの単語が含まれていなくても，これらの単語は適合文書によく関連しており，潜在的にはその文書に現れうる．このように，テキストに現れうる単語の情報があれば，フィードバックと検索結果中の各文書との類似度を算出する際に有用であると考えられる．

そこで，本稿では，テキストに表層的に存在する単語の情報だけでなく，テキストに潜在的に現れうる単語の情報も利用する適合性フィードバックの手法を提案する．提案手法では，まず Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003) を用いて，テキストに潜在するトピックの分布を推定する．次に，推定された潜在トピックの分布を基に，各テキストに潜在的に現れうる単語の分布を推定する．そして，推定された潜在的な単語の分布とテキストの表層的な単語の分布の両方を用いて，フィードバックと検索結果中の各文書との類似度を算出し，これを基に検索結果をリランキングする．実験の結果，2 文書（合計 3,589 単語）から成るフィードバックが与えられたとき，提案手法が初期検索結果の Precision at 10 (P@10) を 27.6% 改善することが示された．また，提案手法が，フィードバックが少ない状況でも，初期検索結果のランキング精度を改善する特性を持つことが示された（e.g., フィードバックに 57 単語しか含まれていなくても，P@10 で 5.3% の改善が見られた）．

以降，本稿では，次の構成に従って議論を進める．2 章では，提案手法の基礎をなす，言語モデルに基づくランキングアルゴリズムについて概説する．3 章では，提案手法で使用する LDA について解説する．4 章では，提案手法について説明する．5 章では，提案手法の有効性を調査するために行った実験と，その結果について報告する．最後に，6 章で，本稿の結論を述べる．

2　言語モデルに基づくランキング

本章では，言語モデルに基づくランキングアルゴリズムについて概説する．ここで紹介する技術は，4 章で説明する提案手法の基礎をなしている．

2.1　概要

言語モデルに基づくランキングアルゴリズムは，三つのタイプに分類できる．すなわち，クエリの尤度に基づく方法 (Ponte and Croft 1998)，文書の尤度に基づく方法 (Lavrenko and Croft 2001)，カルバック・ライブラー情報量に基づく方法 (Lafferty and Zhai 2001) の三つである．

クエリの尤度に基づく方法では，文書セット中の各文書 $\boldsymbol{d}_h$ $(h = 1, \ldots, H)$ について，$\boldsymbol{d}_h$ を表

す言語モデル $P_{\boldsymbol{d}_h}(\cdot)$ を構築する．ユーザによってクエリ $\boldsymbol{q}$ が入力されたら，各文書 $\boldsymbol{d}_h$ について，$P_{\boldsymbol{d}_h}(\cdot)$ がクエリを生成する確率 $P_{\boldsymbol{d}_h}(\boldsymbol{q})$ を計算する．そして，$P_{\boldsymbol{d}_h}(\boldsymbol{q})$ が高い順に各文書をランキングする．

文書の尤度に基づく方法は，クエリの尤度に基づく方法と逆のアプローチを採る．すなわち，クエリ $\boldsymbol{q}$ を表す言語モデル $P_{\boldsymbol{q}}(\cdot)$ を構築し，文書セット中の各文書 $\boldsymbol{d}_h$ について，$P_{\boldsymbol{q}}(\boldsymbol{d}_h)$ を計算する．そして，$P_{\boldsymbol{q}}(\boldsymbol{d}_h)$ が高い順に各文書をランキングする．

カルバック・ライブラー情報量に基づく方法では，$P_{\boldsymbol{q}}(\cdot)$ と $P_{\boldsymbol{d}_h}(\cdot)$ の両方を構築する．そして，各文書 $\boldsymbol{d}_h$ について，$P_{\boldsymbol{q}}(\cdot)$ と $P_{\boldsymbol{d}_h}(\cdot)$ のカルバック・ライブラー情報量 $KL(P_{\boldsymbol{q}}(\cdot)||P_{\boldsymbol{d}_h}(\cdot))$ を計算し，これが小さい順に各文書をランキングする．

2.2 言語モデルの構築方法

クエリや文書を表す言語モデル[1]は，Maximum Likelihood Estimation (MLE) や DIRichlet smoothed estimation (DIR) (Zhai and Lafferty 2004) などの方法を用いて構築する．

MLE では，テキスト $\boldsymbol{t}$（$\boldsymbol{t}$ はクエリや文書）における単語 w の生起確率 $P_{\boldsymbol{t}}^{MLE}(w)$ を次式によって算出する．

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

ただし，$tf(w,\boldsymbol{t})$ は $\boldsymbol{t}$ における w の出現頻度を表す．また，$|\boldsymbol{t}|$ は，$\boldsymbol{t}$ に含まれる単語数を表す．

一方，DIR では，$\boldsymbol{t}$ における w の生起確率 $P_{\boldsymbol{t}}^{DIR}(w)$ を次式によって算出する．

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

ただし，$\boldsymbol{D}_{all}$ は文書セットを表す．また，μ はスムージングパラメータを表す．DIR では，MLE と異なり，$\boldsymbol{D}_{all}$ における w の出現頻度が加味されており，スムージングが行われている．

2.3 代表的な適合性フィードバックの手法

言語モデルに基づくランキングアルゴリズムに対する代表的な適合性フィードバックの手法として，Zhai らの手法 (Zhai and Lafferty 2001) がある．Zhai らの手法では，フィードバックとして与えられた文書集合 $\boldsymbol{F} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_G)$ に対して，$\boldsymbol{F}$ を表す言語モデル $P_{\boldsymbol{F}}(\cdot)$ を構築する[2]．次に，$P_{\boldsymbol{F}}(\cdot)$ と $P_{\boldsymbol{q}}(\cdot)$（初期検索結果を得るために使用したクエリモデル）を足し合わせ，新しいクエリモデルを構築する．そして，新しいクエリモデルを用いて，初期検索結果のランキングを修正する．

[1] 以降，本稿では，クエリを表す言語モデルをクエリモデルと呼ぶ．また，文書を表す言語モデルを文書モデルと呼ぶ．

[2] 以降，本稿では，フィードバックを表す言語モデルをフィードバックモデルと呼ぶ．

Zhai らの手法は，言語モデルに基づくランキングアルゴリズムに対する基本的な適合性フィードバックの手法として重要である．しかし，彼らの手法では，テキストに表層的に存在する単語の情報しか用いられていない．これに対し，提案手法では，テキストに潜在的に現れうる単語の分布を推定し，この情報も用いて適合性フィードバックを行う．

3 LDA

本章では LDA (Blei et al. 2003) について解説する．LDA は，提案手法において，各単語がテキストに潜在的に現れうる確率を推定するために用いられる．

3.1 概要

LDA は文書の生成モデルの一つである．LDA では，文書は複数のトピックから生成されると仮定する．また，文書中の各単語は，各トピックが持つ単語の分布から生成されると仮定する．ある文書における各トピックの混合比 $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ は，$(K-1)$ 単体中の一点を取る．ただし，単体中のある一点が選択される確率は，Dirichlet 分布によって決められるとする．

以上の生成過程をまとめると，LDA における文書 $\boldsymbol{d}$ の生成確率は，次のようにして計算される．

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

ただし，$P(\boldsymbol{\theta}|\boldsymbol{\alpha})$ は，Dirichlet 分布から得られる $\boldsymbol{\theta}$ の生成確率である．$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ は正の実数から構成される K 次元ベクトルで，Dirichlet 分布のパラメータを表す．また，$P(w_j|z_k, \boldsymbol{\beta}_k)$ と $P(z_k|\boldsymbol{\theta})$ は，多項分布から得られる w_j と z_k の生成確率である．z_k $(k = 1, \ldots, K)$ はトピックを，$\boldsymbol{\beta}_k$ は z_k が持つ単語の分布を表す．J は LDA で考慮する語彙数を表す．

3.2 パラメータの推定方法

LDA では，変分ベイズ法やギブスサンプリングなどを用いてパラメータを推定する (Blei et al. 2003; Griffiths and Steyvers 2004)．ギブスサンプリングを用いれば，より厳密な推定結果が得られる．実装も容易なため，一般的にはギブスサンプリングが用いられることが多い．しかし，ギブスサンプリングには推定に時間を要するという欠点がある．一方，変分ベイズ法は，厳密な推定結果は得られないが，高速に動作する．即時性が要求される検索というタスクの性質を考慮し，提案手法では変分ベイズ法を用いる．以下，変分ベイズ法による推定方法について説明する．

まず，訓練データ中の各文書 $\boldsymbol{d}_i$ $(i = 1, \ldots, I)$ について，変分パラメータ $\boldsymbol{\gamma}_i = (\gamma_{i1}, \ldots, \gamma_{iK})$

と $\boldsymbol{\phi}_i = (\boldsymbol{\phi}_{i1}, \ldots, \boldsymbol{\phi}_{iJ})$ を導入する．ただし，$\boldsymbol{\phi}_{ij} = (\phi_{ij1}, \ldots, \phi_{ijK})$ である．そして，式 (4) と式 (5) を交互に計算し，これらの値を更新する．

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

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

ただし，Ψ はディガンマ関数を表す．

次に，更新された $\boldsymbol{\gamma}_i$ と $\boldsymbol{\phi}_i$ を用いて，α_k と $\boldsymbol{\beta}_k$ を更新する．α_k と $\boldsymbol{\beta}_k$ の更新には，ニュートン-ラフソン法や固定点反復法を用いる (Blei et al. 2003; Minka 2000)．ここでは固定点反復法による α_k と $\boldsymbol{\beta}_k$ の更新式を示す．更新式は次の通りである．

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

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

ただし，$n_{ik} = \sum_{j=1}^{J} \phi_{ijk}\ tf(w_j, \boldsymbol{d}_i)$，$\alpha_0 = \sum_{k'=1}^{K} \alpha_{k'}$ とする．また，α_k^{old} は更新前の α_k を表すものとする．

以降，$\boldsymbol{\gamma}_i$ と $\boldsymbol{\phi}_i$ の更新と，α_k と $\boldsymbol{\beta}_k$ の更新を繰り返すことで，各パラメータの値を推定することができる．α_k と $\boldsymbol{\beta}_k$ の値が推定されれば，式 (3) を用いて，文書 $\boldsymbol{d}_i$ の生成確率を求めることができる．また，$\boldsymbol{\gamma}_i$ の値が推定されれば，次式を用いて，文書 $\boldsymbol{d}_i$ における単語 w_j の生起確率 $P_{\boldsymbol{d}_i}^{LDA}(w_j)$ を求めることができる．

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

ここで，$\gamma_{ik} / \sum_{k'=1}^{K} \gamma_{ik'}$ は，$\boldsymbol{d}_i$ に潜在するトピックの分布に相当する．これに基づいて $\boldsymbol{\beta}_{kj}$ を足し合わせることで，w_j が $\boldsymbol{d}_i$ に潜在的に現れうる確率を求めることができる．

3.3 未知テキストに対する適用

LDA は Probabilistic Latent Semantic Analysis (PLSA) (Hofmann 1999) をベイズ的に拡張したモデルと位置付けられる．PLSA に対する LDA の長所として，LDA は未知テキスト（訓練データ中に含まれないテキスト）に関する確率も推定できるという点が挙げられる．未知テキスト $\boldsymbol{t}$ に LDA を適用するときは，$\boldsymbol{t}$ に対して変分パラメータ $\boldsymbol{\gamma}_t$ と $\boldsymbol{\phi}_t$ を導入し，式 (4) と式 (5) を用いてこれらの値を推定する．ただし，α_k と $\boldsymbol{\beta}_k$ には，訓練データによって推定された値を用いる．$\boldsymbol{\gamma}_t$ が推定されれば，式 (8) を用いて，未知テキスト $\boldsymbol{t}$ における単語 w_j の生成確

率 $P_{\boldsymbol{t}}^{LDA}(w_j)$ を求めることができる．提案手法では，LDA のこの長所を利用して，各単語がフィードバックに潜在的に現れうる確率を求めている．

3.4 情報検索における LDA の利用

LDA は，自然言語処理や画像処理，音声認識など，様々な分野で利用されている (Blei et al. 2003; Fei-Fei and Perona 2005; Heidel, an Chang, and shan Lee 2007)．情報検索の分野では，例えば Wei らが，クエリの尤度に基づくランキング手法に LDA を利用している (Wei and Croft 2006)．また，Yi らは文書の尤度に基づくランキング手法に，Zhou らはカルバック・ライブラー情報量に基づくランキング手法に LDA を利用している (Yi and Allan 2009; Zhou and Wade 2009)．これらの研究は，LDA を用いて各文書の文書モデルを構築し，それぞれのスコア（e.g., クエリの尤度）に基づいてクエリに対する検索結果を取得するものである．本研究では，さらに，ユーザからフィードバックが得られる問題（i.e., 適合性フィードバックの問題）に焦点を当てる．我々は，フィードバックに対しても LDA を用いてその言語モデルを構築し，構築されたフィードバックモデルを用いて検索結果を修正する．

4 提案手法

本章では，提案手法の概要と，提案手法を構成する各ステップについて詳説する．

4.1 概要

提案手法では，テキストに表層的に存在する単語の情報だけでなく，テキストに潜在的に現れうる単語の情報も利用して，検索結果をリランキングする．表層情報だけでなく潜在情報も考慮することで，表層的なレベルだけでなく潜在的なレベルでもフィードバックと類似する文書を検索結果の上位にリランキングする．

図 1 に提案手法の概要を示す．以降，本稿では，テキスト $\boldsymbol{t}$ の表層情報と潜在情報の両方を含む言語モデルを $P_{\boldsymbol{t}}^{HYB}(\cdot)$ と表す（HYB は hybrid を表す）．まず，ユーザによって入力されたクエリ $\boldsymbol{q}$ に対して，その初期検索結果 $\boldsymbol{D_q} = (\boldsymbol{d}_1, \ldots, \boldsymbol{d}_I)$ を取得する (**Step 1**)．次に，LDA を用いて，$\boldsymbol{D_q}$ 中の各文書 $\boldsymbol{d}_i$ $(i = 1, \ldots, I)$ について，$\boldsymbol{d}_i$ に潜在的に現れうる単語の分布を推定する．そして，$\boldsymbol{d}_i$ の表層的な単語の分布と潜在的な単語の分布の両方を考慮した言語モデル $P_{\boldsymbol{d}_i}^{HYB}(\cdot)$ を構築する (**Step 2**)．ユーザからフィードバック $\boldsymbol{F} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_G)$ が得られたら，$\boldsymbol{F}$ に対しても LDA を実行し，$\boldsymbol{F}$ に潜在的に現れうる単語の分布を推定する．そして，検索結果中の各文書と同様，$\boldsymbol{F}$ に対しても，$\boldsymbol{F}$ の表層的な単語の分布と潜在的な単語の分布の両方を考慮した言語モデル $P_{\boldsymbol{F}}^{HYB}(\cdot)$ を構築する (**Step 3**)．最後に，構築されたフィードバックモデル $P_{\boldsymbol{F}}^{HYB}(\cdot)$ と，初期検索結果 $\boldsymbol{D_q}$ を得るために使用したクエリモデル $P_{\boldsymbol{q}}^{MLE}(\cdot)$ を混合し，新しいクエリモデル

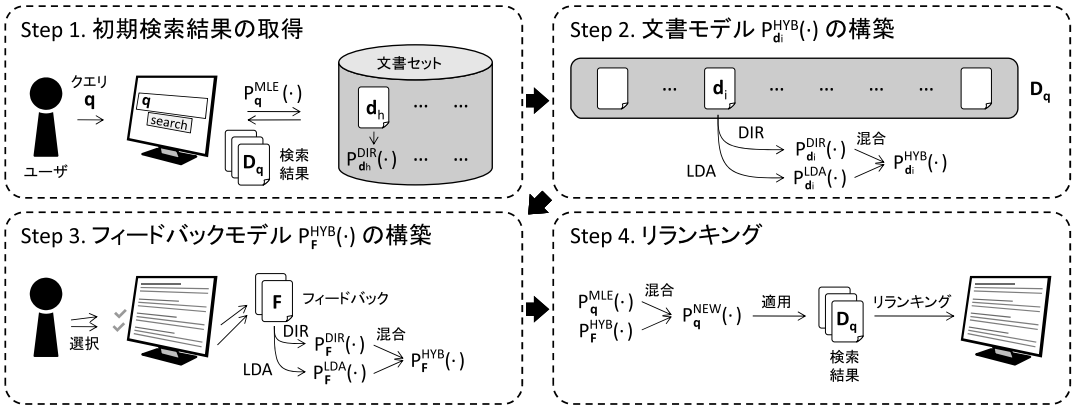

図 1　提案手法の概要

$P_{\boldsymbol{q}}^{NEW}(\cdot)$ を構築する．そして，検索結果中の各文書 $\boldsymbol{d}_i$ について，文書モデル $P_{\boldsymbol{d}_i}^{HYB}(\cdot)$ と新しいクエリモデル $P_{\boldsymbol{q}}^{NEW}(\cdot)$ との類似度を算出し，これに基づいて $\boldsymbol{D_q}$ をリランキングする (**Step 4**)．次節以降では，各ステップについて詳説する．

なお，提案手法とはそもそもの検索モデルが異なるが，テキストの潜在情報を利用するため，Latent Semantic Analysis (LSA) を用いることも考えられる．すなわち，各文書をベクトルで表現し，文書セットに対して LSA を実行する．そして，LSA の実行結果を用いて各ベクトルを低次元の意味的空間に射影することで，各文書に潜在的に現れうる単語の情報を利用することができる．しかし，この方法では，今述べた通り，文書セット全体に対して LSA を実行する必要がある．文書セットは時に数千万〜数億文書にも及ぶため，LSA の実行には膨大な時間を要する．さらに，もし文書セットに対する文書の追加や削除があれば，LSA を実行しなおさなければならない．一方，提案手法では，検索結果中の各文書に対する $P_{\boldsymbol{d}_i}^{LDA}(\cdot)$ やフィードバックに対する $P_{\boldsymbol{F}}^{LDA}(\cdot)$ を構築するため，検索結果に対して LDA を実行する必要がある（4.3 節及び 4.4 節で後述）．しかし，検索結果は文書セットより明らかに規模が小さく，これに要する時間は問題にならない（5.7 節で後述）．このように，LSA に基づく手法と提案手法の間には，ベースとする検索モデルや効率の面で大きな違いがある．

4.2　初期検索結果の取得

提案手法では，カルバック・ライブラー情報量に基づいて (Lafferty and Zhai 2001)，各文書をランキングする．まず，文書セット $\boldsymbol{D}_{all}$ 中の各文書 $\boldsymbol{d}_h$ $(h = 1, \ldots, H)$ について，DIR に基づく文書モデル $P_{\boldsymbol{d}_h}^{DIR}(\cdot)$ をあらかじめ構築しておく．ユーザからクエリ $\boldsymbol{q}$ が与えられると，$\boldsymbol{q}$ に対して MLE に基づくクエリモデル $P_{\boldsymbol{q}}^{MLE}(\cdot)$ を構築する．そして，$\boldsymbol{D}_{all}$ 中の $\boldsymbol{q}$ を含む各文書について，$P_{\boldsymbol{q}}^{MLE}(\cdot)$ と $P_{\boldsymbol{d}_h}^{DIR}(\cdot)$ のカルバック・ライブラー情報量を計算する．すなわち，クエ

リ $\boldsymbol{q}$ に対する文書 $\boldsymbol{d}_h$ の重要度は，次式のように定義される．

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

この重要度に従って各文書をランキングし，$\boldsymbol{q}$ に対する初期検索結果 $\boldsymbol{D_q}$ を得る．

クエリモデルの構築に MLE を用いたのは，言語モデルに基づくランキングに関する先行研究 (e.g., (Zhai and Lafferty 2001)) に倣ってのことである．なお，クエリモデルの構築に MLE を用いた場合，カルバック・ライブラー情報量に基づくランキングは，クエリの尤度に基づくランキング (Ponte and Croft 1998) と等価になる．

4.3　文書モデル $P^{HYB}_{\boldsymbol{d}_i}(\cdot)$ の構築

$\boldsymbol{D_q}$ 中の各文書 $\boldsymbol{d}_i$ $(i = 1, \ldots, I)$ について，$\boldsymbol{d}_i$ の表層情報と潜在情報の両方を含む言語モデル $P^{HYB}_{\boldsymbol{d}_i}(\cdot)$ を構築する．まず，各文書 $\boldsymbol{d}_i$ について，LDA を用いて，$\boldsymbol{d}_i$ の潜在情報を含む言語モデル $P^{LDA}_{\boldsymbol{d}_i}(\cdot)$ を構築する．具体的な手順は次の通りである．まず，$\boldsymbol{D_q}$ に対して LDA を実行し，$\boldsymbol{D_q}$ に対する LDA のパラメータ α_k と $\boldsymbol{\beta}_k$ $(k = 1, \ldots, K)$，$\boldsymbol{\gamma}_i$ $(i = 1, \ldots, I)$ を推定する（3.2 節参照）．次に，各文書について，推定された各パラメータ及び式 (8) を用いて $P^{LDA}_{\boldsymbol{d}_i}(\cdot)$ を構築する．$P^{LDA}_{\boldsymbol{d}_i}(\cdot)$ は，$\boldsymbol{d}_i$ に潜在するトピックの分布を基に構築されており，各単語が $\boldsymbol{d}_i$ に潜在的に現れうる確率の分布になる（式 (8) 参照）．

次に，構築された $P^{LDA}_{\boldsymbol{d}_i}(\cdot)$ と $P^{DIR}_{\boldsymbol{d}_i}(\cdot)$ を次式によって混合し，$P^{HYB}_{\boldsymbol{d}_i}(\cdot)$ を構築する．

$$P^{HYB}_{\boldsymbol{d_i}}(w) = (1-a)P^{DIR}_{\boldsymbol{d_i}}(w) + aP^{LDA}_{\boldsymbol{d_i}}(w) \tag{10}$$

ただし，$0 \leq a \leq 1$ とする．$P^{DIR}_{\boldsymbol{d}_i}(\cdot)$ は，各文書の表層的な単語の分布を基に構築される（式 (2) 参照）．$P^{DIR}_{\boldsymbol{d}_i}(\cdot)$ と $P^{LDA}_{\boldsymbol{d}_i}(\cdot)$ を混合することで，$\boldsymbol{d}_i$ の表層情報と潜在情報の両方を含む言語モデルを構築することができる．

4.4　フィードバックモデル $P^{HYB}_{\boldsymbol{F}}(\cdot)$ の構築

フィードバック $\boldsymbol{F}$ が得られたら，$\boldsymbol{F}$ に対しても，$\boldsymbol{F}$ の表層情報と潜在情報の両方を含む言語モデル $P^{HYB}_{\boldsymbol{F}}(\cdot)$ を構築する．まず，LDA を用いて，$\boldsymbol{F}$ の潜在情報を含む言語モデル $P^{LDA}_{\boldsymbol{F}}(\cdot)$ を構築する．具体的な手順は次の通りである．まず，Step 2 で訓練された LDA を $\boldsymbol{F}$ に適用し，$\boldsymbol{F}$ に対する変分パラメータ $\boldsymbol{\gamma_F}$ を推定する（3.3 節参照）．次に，推定された $\boldsymbol{\gamma_F}$ と式 (8) を用いて $P^{LDA}_{\boldsymbol{F}}(\cdot)$ を構築する．$P^{LDA}_{\boldsymbol{F}}(\cdot)$ は，$P^{LDA}_{\boldsymbol{d}_i}(\cdot)$ と同様，各単語が $\boldsymbol{F}$ に潜在的に現れうる確率の分布になる．

次に，構築された $P^{LDA}_{\boldsymbol{F}}(\cdot)$ と $P^{DIR}_{\boldsymbol{F}}(\cdot)$ を次式によって混合し，$P^{HYB}_{\boldsymbol{F}}(\cdot)$ を構築する．

$$P^{HYB}_{\boldsymbol{F}}(w) = (1-a)P^{DIR}_{\boldsymbol{F}}(w) + aP^{LDA}_{\boldsymbol{F}}(w) \tag{11}$$

ただし，$P_{\boldsymbol{F}}^{DIR}(\cdot)$ は式 (2) を用いて構築する．$P_{\boldsymbol{F}}^{DIR}(\cdot)$ と $P_{\boldsymbol{d}_i}^{LDA}(\cdot)$ を混合することで，$\boldsymbol{F}$ の表層情報と潜在情報の両方を含む言語モデルを構築することができる．

4.5 リランキング

$\boldsymbol{D_q}$ をリランキングするため，まず新しいクエリモデルを構築する．新しいクエリモデル $P_{\boldsymbol{q}}^{NEW}(\cdot)$ は，$\boldsymbol{D_q}$ を得るために使用したクエリモデル $P_{\boldsymbol{q}}^{MLE}(\cdot)$ と，Step 3 で構築したフィードバックモデル $P_{\boldsymbol{F}}^{HYB}(\cdot)$ を次式のようにして混合し，構築する．

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

ただし，$0 \leq b \leq 1$ とする．

最後に，$\boldsymbol{D_q}$ 中の各文書 $\boldsymbol{d}_i$ について，$P_{\boldsymbol{d}_i}^{HYB}(\cdot)$ と $P_{\boldsymbol{q}}^{NEW}(\cdot)$ のカルバック・ライブラー情報量を算出する．すなわち，クエリ $\boldsymbol{q}$ とフィードバック $\boldsymbol{F}$ が与えられた下での文書 $\boldsymbol{d}_i$ の重要度を次式のように定義する．

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

この重要度に従って各文書をリランキングすることで，検索結果のランキングを修正する．

5 実験

本章では，提案手法の有効性を調査するために行った実験と，その結果について報告する．

5.1 実験データ

実験は，第 3 回 NTCIR ワークショップ[3]で構築されたウェブ検索評価用テストセット (Eguchi, Oyama, Ishida, Kuriyama, and Kando 2002) を用いて，これを行った．テストセットは，11,038,720 ページの日本語ウェブ文書と，47 個の検索課題から成る．検索課題ごとに，約 2,000 文書に，その課題に対する適合度が付与されている．ただし，適合度は「高適合」「適合」「部分適合」「不適合」のいずれかである．これらの適合度が付与された文書を用いて，検索結果のランキング精度を測ることができる．

図 2 に検索課題の一例を示す．各タグが表す意味内容は次の通りである．

NUM 課題番号．

TITLE 検索システムに入力するであろう単語．課題作成者によって 2〜3 語がリストアップされている．左から順に重要．

[3] http://research.nii.ac.jp/ntcir/ntcir-ws3/ws-ja.html

⟨NUM⟩ 0042 ⟨/NUM⟩
⟨TITLE⟩ イースター，復活祭 ⟨/TITLE⟩
⟨DESC⟩ キリストの復活を祝う「イースター」の祭りについて書かれている文書を探したい ⟨/DESC⟩
⟨RDOC⟩ NW002542912, NW002008347, NW000837198 ⟨/RDOC⟩

図 2　テストセットにおける検索課題の一例

DESC　課題作成者の情報要求を一文で表したもの．

RDOC　情報要求に適合する代表的な文書の ID．課題作成者によって 2〜3 個がリストアップされている．

実験では，⟨TITLE⟩ タグの単語をクエリとして使用した．ただし，提案手法では，検索の質を高めるため，クエリを含む文書（クエリを構成する各タームが最低でも 1 回以上出現する文書）のみをスコア付けの対象として収集する（4.2 節参照）．そのため，⟨TITLE⟩ タグの全ての単語を用いると，多くの検索課題において，検索される文書数が極端に少なくなってしまった．例えば，課題番号 0027，0047，0058 などは，それぞれ 17 文書，5 文書，14 文書しか検索できなかった．課題番号 0061 に至っては 1 文書も検索できなかった．このように検索される文書が少ないと，適合性フィードバックの有効性が検証しにくい．すなわち，実際に適合性フィードバックによって初期検索結果のランキングが改善されても，その結果が P@10 などの評価尺度の値に反映されにくく，適合性フィードバックが有効に働いたかどうかが判断しづらい．そこで，実験では，この問題を避けるため，十分な検索結果が得られるように，クエリとして使用する単語を ⟨TITLE⟩ タグの最初の 2 語のみとした．ただし，「十分」の定義は「100 文書以上」とした．

また，⟨RDOC⟩ タグの ID が付与された文書を，ユーザのフィードバックとして使用した．上で述べた通り，これらは課題作成者本人によって選択された代表的な適合文書であり，フィードバックとして使用するのに最適と考えられる．これらの文書は，提案手法の初期検索結果に含まれるとは限らない．初期検索結果に含まれない場合，これらをユーザのフィードバックとして使用するのは奇異に感じられるかもしれない．しかし，これらの文書は，仮に初期検索結果に含まれていた場合も，リランキング前後のランキング精度を測定・比較する際，結局ランキングから取り除かれる（5.3 節で後述）．言い換えれば，これらは，初期検索結果に含まれていた場合も，初期検索結果に含まれない場合のように，検索結果中に存在していないものとして扱われる．このように，どちらの場合でも存在していないものとして扱われることを考えると，これらの文書が初期検索結果に含まれているか含まれていないかは重要ではない．以上を踏まえ，実験では，これらが初期検索結果に含まれているか含まれていないかは問題にしなかった．

47 個の検索課題のうち，7 個の検索課題（課題番号: 0011, 0018, 0032, 0040, 0044, 0047, 0061）

については，実験で使用しなかった．これは，上で述べたようにクエリとして使用する単語を2語にしても，十分な文書（i.e., 100 文書）が検索できなかったためである．さらに，残った 40 課題を，開発データと評価データに分けて使用した．開発データは，提案手法のパラメータを最適化するために使用した．評価データは，提案手法のランキング精度を測定するために使用した．開発データには 8 課題（課題番号：0008〜0017）を，評価データには 32 課題（課題番号：0018〜0063）を使用した．

5.2 実験用検索システム

実験を行うため，提案手法に従って適合性フィードバックを行う検索システムを作成した．実装の詳細は以下の通りである．

検索対象とする文書セット (i.e., $\boldsymbol{D}_{all}$) には，テストセットの 11,038,720 文書を使用した．また，文書セット中の各文書について，次の手順に従って文書モデルを構築した．

(1) Shinzato らの手法 (Shinzato, Kawahara, Hashimoto, and Kurohashi 2008) を用いて本文を抽出し，JUMAN (Kurohashi, Nakamura, Matsumoto, and Nagao 1994) を用いて各文を解析する．

(2) 解析結果及び式 (2) を用いて，DIR に基づく文書モデルを構築する．ただし，先行研究 (Zhai and Lafferty 2001; Wei and Croft 2006; Yi and Allan 2009) に倣って，$\mu = 1,000$ とした．

クエリが与えられたら，次の手順に従ってクエリモデルを構築した．

(1) JUMAN を用いてクエリを解析する．

(2) 解析結果及び式 (1) を用いて，MLE に基づくクエリモデルを構築する．

LDA の実装については次の通りである．パラメータ α_k $(k = 1, \ldots, K)$ の初期値は 1 とした．また，$\boldsymbol{\beta}_k$ $(k = 1, \ldots, K)$ の初期値にはランダムな値を与えた．$\boldsymbol{\gamma}_i$ と $\boldsymbol{\phi}_i$ を更新する際の反復回数と，α_k と $\boldsymbol{\beta}_k$ を更新する際の反復回数は，それぞれ 10 回とした．LDA で考慮する語彙数 J は 100 とした．ただし，LDA で考慮する語彙は，初期検索結果に対する重要度を基に選出した．ここで，初期検索結果 $\boldsymbol{D_q}$ に対する単語 w の重要度は，$df(w, \boldsymbol{D_q}) * \log(H/df(w, \boldsymbol{D}_{all}))$ と定義した．ただし，$df(w, \boldsymbol{D})$ は $\boldsymbol{D}$ における w の文書頻度を表す．

5.3 ランキング精度の測定方法

適合性フィードバックの効果は，適合性フィードバック前のランキング（i.e., 初期検索結果のランキング）と，適合性フィードバック後のランキングを比較することで検証できる．このとき，フィードバックとして使用する文書の扱いに気を付けなければならない (Hull 1993)．

例えば，適合性フィードバック前後のランキングをそのまま比較すると，後者が有利になってしまう．これは，フィードバックとして与えられた文書（適合であることが分かっている文

書）が，適合性フィードバック後のランキングの上位に含まれやすいためである．

そこで，適合性フィードバック前後のランキングを比較する際，フィードバックとして与えられた文書を適合性フィードバック後のランキングから取り除くという方法が考えられる．しかし，この方法だと，適合性フィードバック前のランキングが有利になってしまう．これは，適合文書が少ないときに特に問題となる．

以上を踏まえ，実験では，ランキングの精度を測定する際，フィードバックとして使用した文書を各ランキングから取り除いた．これにより，適合性フィードバック前後のランキングを公平に比較することができる．

ランキング精度の評価尺度には，P@10，Mean Average Precision (MAP)，Normalized Discounted Cumulative Gain at 10 (NDCG@10) (Järvelin and Kekäläinen 2002) を用いた．ただし，P@10 及び MAP を測定する際は，「高適合」「適合」「部分適合」の文書を正解，「不適合」及び適合度が付与されていない文書を不正解とした．また，NDCG@10 は，「高適合」の文書を 3 点，「適合」の文書を 2 点，「部分適合」の文書を 1 点として算出した．

5.4　リランキング性能の調査

まず，提案手法が初期検索結果のランキング精度をどの程度改善できるか調査した．具体的には，初期検索結果のランキング精度と，提案手法によってリランキングを行った後のランキング精度を比較し，提案手法の有効性を検証した．実験には評価データを使用し，各検索課題の初期検索結果を取得する際は，5.1 節で述べたように，⟨TITLE⟩ タグの最初の 2 単語をクエリとして用いた．また，実験では，*initial_score*（式 (9) 参照）の上位 100 件を初期検索結果とした．提案手法を実行する際は，⟨RDOC⟩ タグの最初の 2 文書をフィードバックとして用いた．なお，これらの文書に含まれる単語数は平均 3,589 語であった．提案手法に必要な 3 つのパラメータ a，b，K の値は，それぞれ 0.2，0.9，50 とした．これらは，5.6 節で述べる実験の結果を基に決定した．

結果を表 1(a) に示す．INIT は各検索課題に対する初期検索結果のランキング精度の平均値

表 1　リランキング性能の調査結果

(a) 適合性フィードバックに対する性能

	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) 擬似適合性フィードバックに対する性能

	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

を，OURS は提案手法実行後のランキング精度の平均値を表す．比較のため，初期検索結果に対してベースラインとなる手法を実行したときの結果も示した．ZHAI は Zhai らの手法 (Zhai and Lafferty 2001) を，OURS ($a = 0.0$) は提案手法から潜在情報を除いた手法を表す．ただし，ZHAI と OURS ($a = 0.0$) は本質的にはほとんど同じ手法である．両手法とも，フィードバックの表層の単語分布を文書セット全体の単語分布で補正することでフィードバックモデルを構築し，これを用いてリランキングを行っている．違うのは単語分布の補正の仕方だけである（前者は EM アルゴリズムを用い，後者は DIR を用いて補正を行っている）．OURS ($a = 0.0$) では，$b = 0.5$ とした．これも，5.6 節で述べる実験の結果を基に決定した．

DIC もベースラインとなる手法を表す．提案手法の核となるアイディアは，テキスト（フィードバック及び検索結果中の各文書）に潜在的に現れうる単語の情報を適合性フィードバックに利用することである．同義語辞書や関連語辞書などの知識リソースを用いても，同様のアイディアを実現することができる．DIC では，OURS ($a = 0.0$) をベースに，テキスト中の各単語が同義語を持つ場合，その同義語もそのテキストに出現しているとみなした上でリランキングを行った．ただし，同義知識は，Shibata らの手法 (Shibata, Odani, Harashima, Oonishi, and Kurohashi 2008) を用いて例会小学国語辞典 (田近 2001) と岩波国語辞典 (西尾，岩淵，水谷 2002) から獲得した．獲得された同義知識（e.g.,「コンピュータ」=「電子計算機」,「ポテト」=「じゃが芋」=「ばれいしょ」）は 4,597 個であった．

表 1(a) を見ると，すべての尺度において，OURS が INIT を大きく上回っている．例えば P@10 は 27.6% 改善しており，提案手法が初期検索結果をうまくリランキングできたことが分かる．また，提案手法は，ZHAI や OURS ($a = 0.0$) より高い性能を示した．ZHAI や OURS ($a = 0.0$) は，テキストの表層情報だけを用いて適合性フィードバックを行っている．一方，提案手法は，テキストの表層情報に加え，テキストの潜在情報も用いて適合性フィードバックを行っている．提案手法がこれらの手法を上回ったことから，潜在情報が適合性フィードバックに有用であったことが分かる．

さらに，リランキング結果を調査したところ，提案手法が，テキストに表層的には出現しないが潜在的には現れうる単語の情報をうまく利用していることが確認できた．図 2 の検索課題を例に取ると，「宗教」や「祝日」「聖書」などの単語は，情報要求によく関連するが，フィードバックとして使用した文書には含まれていなかった．そのため，ZHAI や OURS ($a = 0.0$) では，これらの単語の情報を使用することができなかった．一方，提案手法では，これらの単語がフィードバックにおいてもある程度の確率で現れうると推定できた．具体的には，「宗教」「祝日」「聖書」は，それぞれ 0.0046，0.0037，0.0024 の確率で現れうると推定できた．なお，フィードバックに 1 回出現した単語として「クリスマス」や「ＥＡＳＴＥＲ」などがあったが，これらの生起確率の推定値は，それぞれ 0.0093，0.0060 であった．提案手法では，これらの推定結果を用いることで，これらの単語を含む検索結果中の適合文書を上位にリランキングすること

ができた．

DIC はあまり有効に機能せず，その結果は ZHAI や OURS ($a = 0.0$) の結果を少し上回る程度であった．この原因は，我々が構築した同義語辞書のカバレッジにあると思われる．DIC は，よりカバレッジの高い知識リソースが利用できれば（同義語や関連語などの知識をより多く利用できれば），より有効に機能する可能性を持つ．しかし，そのようなリソースを構築するのは容易ではない．一方，提案手法でも，単語と単語が関連するという知識を必要とする．しかし，DIC と違って，何のリソースも必要としない．すなわち，提案手法では，LDA を用いることで，単語と単語が関連するという知識を検索結果から動的に獲得することができる．1 章の「マック␣価格」というクエリを例に取ると，このクエリに対する検索結果には「CPU」や「ハードディスク」「ハンバーガー」「ポテト」などの単語が含まれると考えられる．提案手法では，検索結果に対して LDA を実行することで，「CPU」と「ハードディスク」が関連するという知識や「ハンバーガー」と「ポテト」が関連するという知識を，トピックという形で動的に獲得することができる．そして，獲得された知識を用いることで，文書に「ハードディスク」という単語が出現していなくても，「CPU」という単語が出現していれば，「ハードディスク」も潜在的にはその文書に現れうると推測できる．このように，DIC と比べると，（カバレッジの高低に関わらず）何のリソースも必要としないという点で，提案手法の方が優れている．

提案手法は擬似適合性フィードバックにも適用可能である．そこで，これに対するリランキング性能も調査した．擬似適合性フィードバックでは，初期検索結果の上位 n 文書を適合文書とみなし，適合性フィードバックを行う．実験では，$n = 10$ として初期検索結果をリランキングし，リランキング前後のランキング精度を比較した．ただし，擬似適合性フィードバックでは，明示的なフィードバック（適合であることが分かっている文書）は存在しない．そのため，ランキングの精度を測る際，他の実験のように，⟨RDOC⟩ タグの文書を各ランキングから除くことはしなかった．

結果を表 1(b) に示す．INIT の値が表 1(a) と違うのは，ランキング精度を算出する際，⟨RDOC⟩ タグの文書を除いていないからである．表 1(b) を見ると，普通の適合性フィードバックに比べると改善の度合いは小さいが，P@10 や NDCG@10 の値が上昇している．例えば，P@10 では 8.2% の改善が見られる．このことから，擬似適合性フィードバックにおいても提案手法がある程度機能することが分かる．

5.5　フィードバックが少ない状況でのリランキング性能

現実的には，ユーザが多くのフィードバックを与えてくれるとは考えにくい．そのため，適合性フィードバックの手法は，フィードバックが少ない状況でも機能するべきである．この実験では，このような状況をシミュレートし，フィードバックが少なくても提案手法が機能するかを調査した．具体的には，提案手法に与えるフィードバックを少しずつ減らしていき，リラ

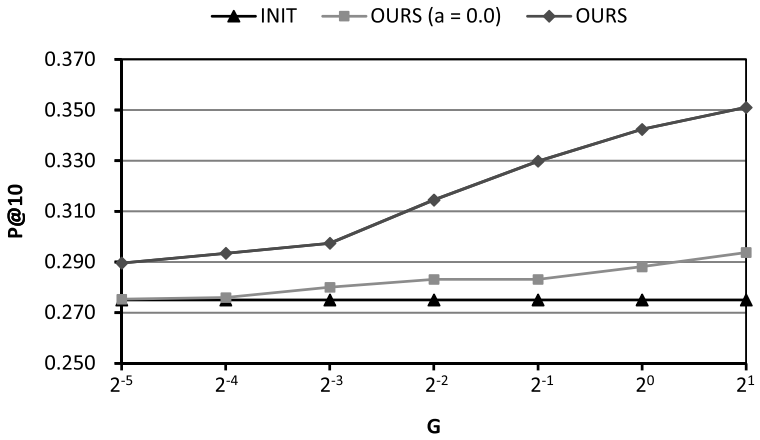

図 3 G によるリランキング性能の変化

ンキング性能がどのように変化するかを調査した．提案手法に与えるフィードバックの分量 G は，$G = 2^1, 2^0, 2^{-1}, \ldots, 2^{-5}$ とした．ただし，例えば $G = 2^1$ は，フィードバックとして 2 文書を用いることを意味している．また，例えば $G = 2^{-1}$ は，フィードバックとして 1 適合文書の半分だけを用いることを意味している．この場合，適合文書中の単語をランダムに半分抽出し，それらを用いて適合性フィードバックを行った．$G < 1$ の場合も調査したのは，フィードバックとして文書より小さい単位（e.g., 文書のタイトル，スニペット）が与えられた場合を想定し，このような場合にも提案手法が機能するかを調べたかったからである．

結果を図 3 に示す．比較のため，提案手法から潜在情報を除いたとき (i.e., OURS ($a = 0.0$)) の性能の変化も示した．また，INIT は初期検索結果のランキング精度を表す．図から，G が小さいときでも，提案手法が高い性能を示すことが分かる．例えば $G = 2^0$ のとき，提案手法は初期検索結果を 24.5% 改善している．さらに，$G = 2^{-5}$ のときでも，5.3% の改善が見られた．なお，$G = 2^{-5}$ のとき，フィードバック $\boldsymbol{F}$ に含まれる単語数は平均 57 語であった．一方，OURS ($a = 0.0$) を見ると，G が小さくなるにつれ，ほとんど改善が見られなくなった．OURS ($a = 0.0$) ではテキストの表層情報しか利用していない．そのため，G が小さくなるにつれて利用できる情報が少なくなり，初期検索結果を改善できなくなったと考えられる．一方，提案手法では，表層情報だけでなく潜在情報も利用している．利用できる情報が多い分，G が小さいときでも，初期検索結果のランキングを改善することができたと考えられる．

5.6 パラメータとリランキング性能の関係

提案手法には 3 つのパラメータ a，b，K がある．a は $P^{DIR}_{\boldsymbol{d}_i}(\cdot)$ と $P^{LDA}_{\boldsymbol{d}_i}(\cdot)$ の混合比を調整するパラメータ（式 (10) 及び式 (11) 参照），b は $P^{MLE}_{\boldsymbol{q}}(\cdot)$ と $P^{HYB}_{\boldsymbol{F}}(\cdot)$ の混合比を調整するパラ

メータ（式 (12) 参照），K は LDA のトピック数である．5.4 節及び 5.5 節で述べた実験では，OURS のパラメータを $a = 0.2$，$b = 0.9$，$K = 50$ とした．また，OURS ($a = 0.0$) のパラメータを $b = 0.5$ とした．これらの値は予備実験の結果を基に決定した．

提案手法の性能を最大限に発揮するためには，パラメータとリランキング性能の関係について知る必要がある．予備実験では，この関係を知るため，様々な (a, b, K) の組み合わせについて提案手法のリランキング性能を調査し，その結果を比較した．ただし，$a = 0.0, 0.1, \ldots, 1.0$，$b = 0.0, 0.1, \ldots, 1.0$，$K = 10, 20, \ldots, 100$ とし，全 1,210 通りの組み合わせについて，調査を行った．開発データを用いて調査した．

ある (a, b, K) の組み合わせに対するリランキング性能は，他の実験と同じようにして，これを測定した．すなわち，開発データ中の各検索課題について初期検索結果を取得し，提案手法を用いてこれらをリランキングした後，全課題における P@10 の平均値を算出した．他の実験と同様，クエリには ⟨TITLE⟩ タグの最初の 2 単語を，フィードバックには ⟨RDOC⟩ タグの最初の 2 文書を用いた．

結果を表 2 及び図 4 に示す．表 2 は，実験結果を (a, b) についてまとめたものである．表中の各セルの値は，各 (a, b) の組み合わせについて，各 K の P@10 を平均したものである．例えば，$(a, b) = (0.1, 0.2)$ のセルは，$(a, b, K) = (0.1, 0.2, 10), (0.1, 0.2, 20), \ldots, (0.1, 0.2, 100)$ の P@10 の平均値が 0.286 であったことを示している．各列においてもっとも P@10 が高いセルは，その値を太字で装飾した．また，各行においてもっとも P@10 が高いセルは，その値に下線を引いた．

表から，$(a, b) = (0.1, 0.9)$ or $(0.2, 0.9)$ のとき，リランキング性能がもっとも良いことが分かる．また，$a = 0.0$ のとき（潜在情報を考慮しないとき）は，b が大体 0.3〜0.5 のとき，リラン

表 2　(a, b) とリランキング性能の関係

		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

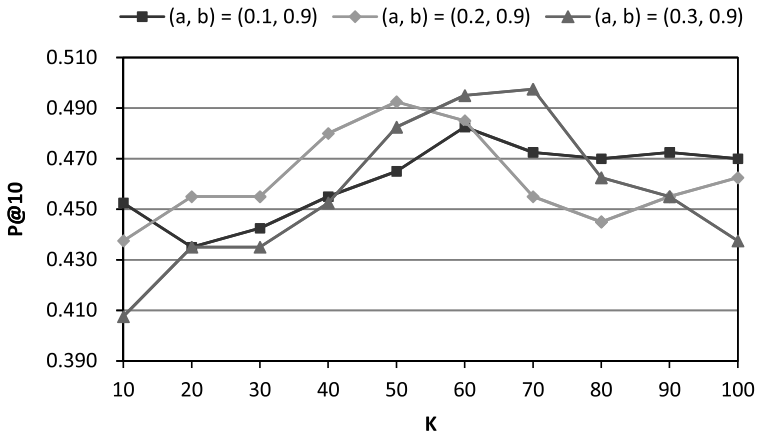

図 4 K によるリランキング性能の変化

キング性能が良い．一方，$a \geq 0.1$ のとき（潜在情報を考慮したとき）は，b が大体 0.8〜1.0 のとき，リランキング性能が良い．$a = 0.0$ のときより，性能が良くなる b の値（及びそのときのランキング精度）が大きくなっている．これは，潜在情報を考慮することで，フィードバックモデルの信頼度が増すことを示唆している．

図 4 は，K によるリランキング性能の変化を示している．図では，表 2 においてリランキング性能が良かった 3 つの (a, b) の組み合わせ $(a, b) = (0.1, 0.9), (0.2, 0.9), (0.3, 0.9)$ について，K による性能の変化を示した．図から，K が大体 50〜70 のとき，リランキング性能が良いことが分かる．

以上の結果をまとめると，提案手法がその性能を発揮するパラメータは，$(a, b) = (0.1, 0.9)$ or $(0.2, 0.9)$，K は大体 50〜70 となる．

5.7 LDA の実行時間

提案手法では，検索結果中の各文書に対する $P_{\boldsymbol{d}_i}^{LDA}(\cdot)$ を構築するため，検索結果に対して LDA を実行する．また，フィードバックに対する $P_{\boldsymbol{F}}^{LDA}(\cdot)$ を構築する際は，フィードバックに対して LDA を実行する．本節では，これらの処理に要する時間について考察する．

実験では，各検索課題の検索結果（100 文書）に対して LDA（Perl と C を組み合わせて実装）を実行するのに，13.1〜16.0 秒を要した．この程度の時間であれば，提案手法を実行する上で，問題にはならない．適合性フィードバックは，(1) システムによる検索結果の提示，(2) ユーザによる検索結果の閲覧，適合文書の選択，(3) 適合文書を用いた検索結果のリランキングという三つのステップから成る．ここで，一般的に考えて，(2) には 1 分以上はかかると思われる．従って，まずユーザに検索結果を提示し，ユーザが検索結果を閲覧している裏で LDA を実行するようなシステムの構成を採れば，(3) に移る前に LDA の実行を終えることができる．こ

のように，検索結果が 100 文書程度であれば，LDA の実行時間は問題にならない．

一方，検索結果は，より大きくなり得る．検索結果が大きくなると，LDA の実行時間も大きくなってしまう．これを解決する一つの方法は，ランキングの上位だけを検索結果とすることである．例えば，多くの文書が検索されても，上位 100 文書だけを検索結果とすれば，上述の通り，LDA の実行時間は問題にならない．別の方法として，変分パラメータの推定を並列化することも考えられる．LDA の実行時間は，変分パラメータの推定に要する時間が多くを占める．ここで，各文書に対する変分パラメータは，他の文書に対する変分パラメータと独立である．従って，各文書に対する変分パラメータの推定を並列化し，LDA の実行時間を削減することができる．例えば，Nallapati らは，50 ノードのクラスタを用いることで LDA の実行時間を 14.5 倍高速化できたと報告している (Nallapati, Cohen, and Lafferty 2007)．提案手法でも並列化を取り入れることで，LDA の実行時間を削減することができると思われる．

最後に，フィードバックに対して LDA を実行するのに要した時間を報告する．これは 1 秒にも満たないものであった．例えば，フィードバックが 2 文書の場合，実行に要した時間は，わずか 0.1〜0.2 秒であった．従って，フィードバックに対する LDA の実行時間も問題にはならない．

6　おわりに

本稿では，テキストの表層情報と潜在情報の両方を利用する適合性フィードバックの手法を提案し，その有効性について議論した．提案手法では，LDA を用いて，フィードバックや検索結果中の各文書に潜在的に現れうる単語の分布を推定した．そして，表層的な単語の分布と潜在的な単語の分布の両方を用いてフィードバックと検索結果中の各文書との類似度を算出し，これに基づいて検索結果をリランキングした．実験では，2 文書（合計約 3,589 単語）から成るフィードバックが与えられたとき，提案手法が初期検索結果の P@10 を 27.6% 改善することを示した．また，提案手法が，フィードバックが少ない状況でも，初期検索結果のランキング精度を改善する特性を持つことを示した（e.g., フィードバックに 57 単語しか含まれていなくても，P@10 で 5.3% の改善が見られた）．

今後の課題としては，ネガティブフィードバックの利用が挙げられる．提案手法は高い性能を示したが，ポジティブフィードバック（ユーザが適合と判定した文書）を扱う機構しか持ち合わせていない．ネガティブフィードバック（ユーザが不適合と判定した文書）も利用することで，さらに性能を上げることができないか検討中である．

参考文献

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., an Chang, H., and shan 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.

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.

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.

西尾実，岩淵悦太郎，水谷静夫 (2002). 岩波国語辞典. 岩波書店.

田近洵一 (2001). 例解小学国語辞典. 三省堂.

略歴

原島　純：2007 年京都大学工学部電気電子工学科卒業．2009 年同大学院情報学研究科修士課程修了．現在，同大学院博士後期課程在学中．情報検索の研究に従事.

黒橋　禎夫：1989 年京都大学工学部電気工学第二学科卒業．1994 年同大学院博士課程修了．京都大学工学部助手，京都大学大学院情報学研究科講師，東京大学大学院情報理工学系研究科助教授を経て，2006 年京都大学大学院情報学研究科教授，現在に至る．自然言語処理，知識情報処理の研究に従事.

（2011 年 12 月 2 日　受付）
（2012 年 3 月 6 日　再受付）
（2012 年 5 月 14 日　採録）