MineBlog：興味発見を支援するblog記事推薦システム

森本和伸[†,☆] 林 貴宏[††] 尾内理紀夫[†,††]

本稿では，ユーザが書いたblog記事をもとに，ユーザの新たな興味につながる可能性のある他人のblog記事を推薦するシステムMineBlogについて述べる．MineBlogでは，あらかじめウェブクローラを利用してblog記事をウェブから収集しデータベース化しておく．データベース中のblog記事から推薦記事を決定するために，関連性，相違性，話題性の3つの尺度を用いてblog記事を評価する．3つの尺度は順に，ユーザが書いた記事とどの程度関連する話題を含んでいるかを測る尺度，ユーザが書いた記事とどの程度異なる話題を含んでいるかを測る尺度，一時期頻繁に話題にされた内容を記事中にどの程度含んでいるかを測る尺度である．これら3つの尺度を定量化し，関連度，相違度，話題度を定義する．3つの尺度によりblog記事をスコアリングし，上位を推薦記事としてユーザに提示する．MineBlogの有効性評価を目的とした実験により，推薦した記事の約2つに1つはユーザの新たな興味につながる推薦記事であるという結果を得た．

MineBlog: A System for Arousing Interest by Recommending Blog Articles

Kazunobu Morimoto,[†,☆] Takahiro Hayashi[††] and Rikio Onai[†,††]

This paper reports on MineBlog: a recommender system of blogs. Posting a blog-article to the system, the system produces attractive blog-articles to the user and supports the user to discover his/her new interests. The system extracts some blog-articles for recommendation from a database by scoring with three kinds of criteria — relevance, difference and topicality. Relevance is a criterion for measuring similarity between an article registered in a database and the user's posted article. Difference is a criterion for measuring dissimilarity between the articles. Topicality is a criterion for measuring whether an article mentions current topics. We experimentally examine the performance of MineBlog. Experimental results show that one of two recommended articles arouse user's new interests.

1. はじめに

blogと呼ばれるウェブページが注目されている．blogとは特定の書き手によって頻繁に更新され，更新された順に文章が並べられているウェブページのことである[1)]．大半のblogは日記のように個人的な内容が記述されたものである[2)]．また，個人がblogを書く動機は自分の記録を残したい，解説や意見を提供したい，アイデアを言葉にしておきたい，感情を表現したいといった個人的な理由によるものが多いため，blog内の文章は記述した人の個性，感情，視点が強く表れたものとなっている[3)]．このように，blogは主観的に様々な内容が記述されているという特徴を持つため，自分とは異なった視点や感覚の文章が書かれたblogを読むことにより，新たな興味を喚起されることがある．しかし，このようなblogを意図的に発見することは困難である．キーワードを用いたblog検索システム☆☆が開発されているが，新たな興味が未知である段階においてキーワードを決定することは困難であり，キーワード検索では新たな興味が喚起されるblogを検索することは難しい．

キーワードを用いない検索手法として，文章をキーとして類似する文章を検索する手法がある．この手法では，キー文章と類似する文章は，キー文章と関連している内容だけでなく少なからず異なる内容も含んで

† 電気通信大学大学院電気通信学研究科
Graduate School of Electro-Communications, The University of Electro-Communications
†† 電気通信大学電気通信学部
Department of Computer Science, The University of Electro-Communications
☆ 現在，富士通株式会社
Presently with Fujitsu Limited

☆☆ Bulkfeeds, http://bulkfeeds.net/
Blogdex, http://blogdex.net/
未来検索 livedoor, http://sf.livedoor.com/
もぶろげっと，http://mobloget.jp/

いる可能性がある．よって，類似文章をユーザに推薦することでユーザにとって未知なものを検索できる可能性がある．しかし，blog は簡単な感想を述べただけの短い文章が多いため，キー文章と関連している内容を含んでいるだけでは異なる内容を含んでいる可能性は低いと考えられる．したがって，キー文章をもとに類似文章を検索する手法をそのまま興味発見へと応用することは難しい．blog の文章を利用して興味発見へとつなげていくためには，キー文章との関連性を評価するだけでなく，相違性をも明示的に評価する必要があると考えられる．これにより，関連性が高く，かつ，相違性も含まれる文章を推薦することができ，新たな興味の発見へとつながる可能性が出てくる．

もう 1 つ新たな興味につながる文章を検索する手法として，多数の blog ライタたちの間で話題になった内容を検出し，ユーザに提示する手法が考えられる．もし多数のライタたちによって話題になっているにもかかわらず，ユーザがその内容について未知だった場合，この話題は他のライタ同様ユーザの興味を引く可能性は高い．blog の記事には作成日時が付加されており，これを利用することで話題の新しさが判定できる．その結果，最近話題になっている記事などが推薦可能である．

このように，ユーザが書いた文章と関連性と相違性が高く，かつ，話題性も高い文章をユーザに提示することで，ユーザが新たな興味の発見へとつながる可能性が出てくると考える．

そこで本稿では，関連性，相違性，話題性を考慮し，ユーザが書いた blog 記事を用いて新たな興味の発見につながる blog 記事を推薦する手法を提案し，その手法を用いた blog 記事推薦システム MineBlog を実装し，実験により評価を行った．

以下，2 章でシステムの概要を説明する．3 章では推薦記事の決定法について述べる．4 章で MineBlog の有効性を評価するために行った実験について述べる．5 章で関連研究について述べる．6 章で本稿をまとめる．

なお，本稿で使用する用語の意味は以下のとおりである．

blog システム：blog ページを作成，管理するシステム．MovableType ☆1，tDiary ☆2，ココログ☆3，はてなダイアリー☆4など．

blog ページ：blog システムによって作成されたウェブページ．

blog 記事：blog ページにおいて，1 日分，もしくは 1 投稿分の文章．また，本稿で単に記事といえば blog 記事を指す．

2. システム概要

MineBlog では，ユーザに対して blog 記事を書いてもらい，その記事を用いてウェブよりあらかじめ集めておいた記事の中から推薦記事を決定する．本章ではこのシステムの概要について述べる．

MineBlog のシステム構成を**図 1** に示す．処理はウェブより blog を収集しデータベースに格納する処理（図 1 一点鎖線内，以下収集処理と呼ぶ），ユーザが書いた記事より推薦記事を決定する処理（図 1 破線内，以下推薦処理と呼ぶ）に分かれる．

収集処理は以下の手順で行う（以下の箇条書きの番号は，図 1 の番号 (1)〜(4) に対応している）．

(1) ウェブクローラを利用して blog ページを収集する．
(2) blog 記事抽出部で blog ページを記事単位に分割する．
(3) メタデータ抽出部で各記事に対し，メタデータとして記事中に存在する単語（MeCab ☆5で抽出）と記事の書かれた日付を取り出す．
(4) 記事とそれに付随するメタデータをデータベー

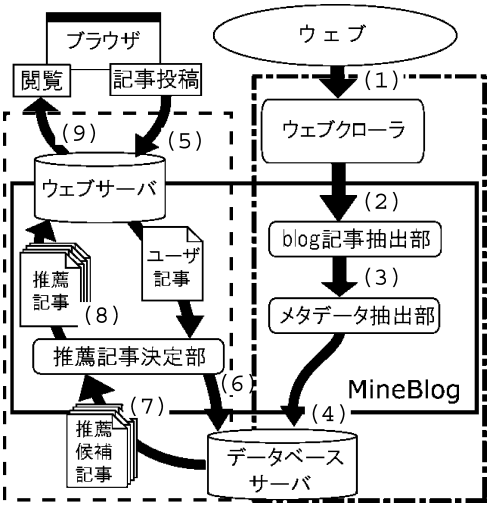


図 1 システム構成
Fig. 1 Structure of a system.

☆1 MovableType, http://www.sixapart.com/movabletype/
☆2 tDiary, http://www.tdiary.org/
☆3 ココログ, http://www.cocolog-nifty.com/
☆4 はてなダイアリー, http://d.hatena.ne.jp/
☆5 MeCab, http://chasen.org/~taku/software/mecab/

表 1 収集対象とした blog サービス
Table 1 Blog services from which MineBlog has retrieved blog pages.

サービス名	URL
ブログ人	http://blog.ocn.ne.jp/
ココログ	http://www.cocolog-nifty.com/
Doblog	http://www.doblog.com/
excite ブログ	http://www.exblog.jp/
はてなダイアリー	http://d.hatena.ne.jp/
JUGEM	http://jugem.jp/
livedoor Blog	http://blog.livedoor.com/
melma!blog	http://blog.melma.com/
Seesaa ブログ	http://blog.melma.com/
yaplog!	http://www.yaplog.jp/

スに格納する．

なお，ウェブ全体から blog ページを判定収集し，さらに記事を抽出するという処理の実現は非常に困難である．そこで，著者らが設計，実装した MineBlog では，blog ページの収集方法として，**表 1** に示した特定の blog サービスのウェブサイトにある blog ページのみを収集することにしている．これら blog サービス利用者の記事の一覧ページが各 blog サービスにより提供されているので，この情報をもとにクローラの収集先を決定している．また，blog ページの収集戦略として，1 ユーザの記事をすべて収集するよりは，多くのユーザの記事を幅広く収集したほうが内容的に偏りの少ない記事収集ができると考え，幅優先探索を採用している．収集先を特定の blog サービスのウェブサイトに限定しているため，blog ページからの記事抽出では，各 blog サービス固有の HTML のフォーマット（レイアウト）が利用可能である．MineBlog ではあらかじめ各 blog サービスごとに HTML マークアップの規則をルール化しておき，これを利用して記事や日付抽出を行っている．2005 年 3 月現在，約 120 万の blog 記事がデータベースに格納されている．

推薦処理は以下の手順で行う（以下の箇条書きの番号は，図 1 の番号 (5)〜(9) に対応している）．

(5) ユーザが投稿した記事をウェブサーバより受け取る．
(6) ユーザが投稿した記事から特徴語を抽出しデータベースサーバに送信する．
(7) データベースより推薦候補記事を取り出す．
(8) 推薦記事決定部で推薦記事を決定しウェブサーバに送信する．
(9) **図 2** のように，ユーザの投稿記事の下に推薦記事を表示し，ユーザは推薦記事を閲覧をする．

以上のように収集処理，推薦処理を行うことで，ユーザに推薦記事を提示する．

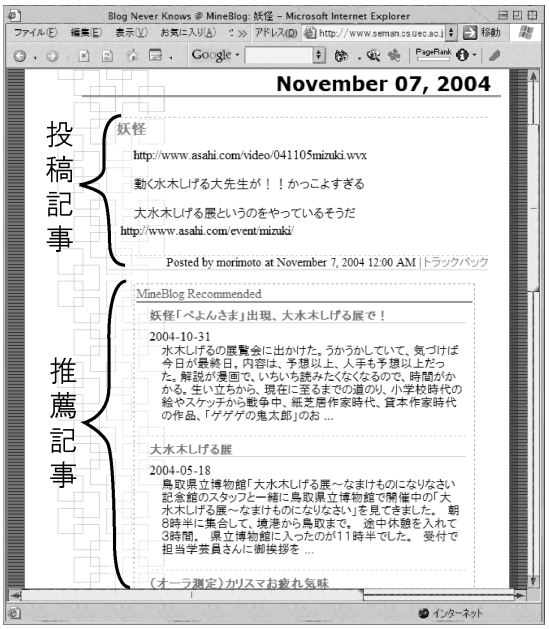


図 2 推薦例
Fig. 2 Example of recommendation.

3. 推薦記事決定法

本章では推薦記事決定部における推薦記事決定法について詳しく述べる．

3.1 推薦記事決定部の処理の流れ

まず，推薦記事決定部での処理の流れは以下の手順で行う（以下の箇条書きの番号は，**図 3** の番号に対応している）．

(1) ユーザが投稿した記事から特徴語を抽出する．
(2) ユーザが投稿した記事に含まれる特徴語と同じ単語を 2 単語以上含むデータベース中の記事を推薦候補記事として取り出す．
(3) この取り出した各推薦候補記事から特徴語を抽出する．
(4) 取り出した推薦候補記事のうちノイズとなる記事は除去する．
(5) ユーザが投稿した記事と推薦候補記事とを特徴語を用いて比較し，各推薦候補記事を関連性，相違性，話題性により評価しスコアリングする．
(6) スコアリングにより上位になった推薦候補記事を推薦記事と決定しウェブサーバに送信する．

推薦記事を決定するために必要となる特徴語の抽出，ノイズ記事の除去，スコアリングの各処理についてそれぞれ 3.2 節，3.3 節，3.4 節で詳しく述べる．

3.2 特徴語抽出

特徴語とは，対象の記事中に多く含まれるが他の記

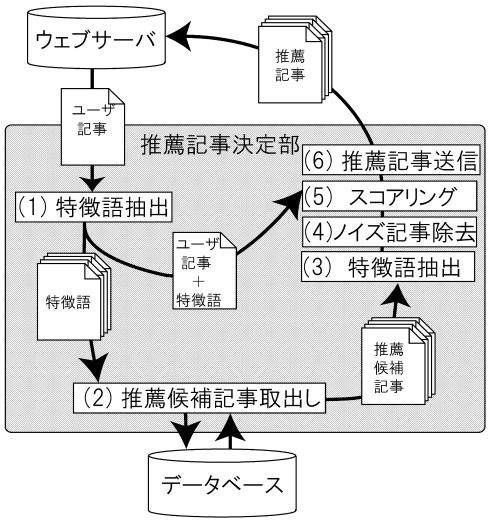


図 3 推薦記事決定部の流れ
Fig. 3 Flow of module of deciding recommendation article.

事にはあまり登場しない単語である．このような単語は記事を特徴付ける重要な単語であるということが知られている[4]．本節では blog 記事からこの特徴語を抽出するための手法について述べる．

3.2.1 単語連結処理

特徴語抽出の前処理である単語連結処理について述べる．まず対象となる記事の文章を形態素解析し，文章から単語とその品詞を抽出する．MineBlog の実装ではこの形態素解析には MeCab を利用している．形態素解析により抽出される単語は分解可能な最小の単語となる．たとえば「新潟中越地震の被害は甚大であった。」という文を形態素解析すると次のように分解される．

新潟（名詞）/ 中越（名詞）/ 地震（名詞）/ の（助詞）/ 被害（名詞）/ は（助詞）/ 甚大（名詞）/ で（助動詞）/ あっ（助動詞）/ た（助動詞）/ 。（記号）
（括弧内は単語の品詞）

形態素解析（MeCab）の結果は「新潟」「中越」「地震」と分割されるが，「新潟中越地震」と連続する名詞を連結して 1 単語として扱うことで記事中の内容を表す特徴的な単語となる．もし，これらの 3 つの単語を連結せず独立して扱った場合，それぞれの単語がカバーする意味の範囲が広くなりすぎ，関連性や相違性を求める際に問題になると考えられる．

また，blog 記事には，新語や固有名詞などが多く出現する．形態素解析において，新語や固有名詞の多くが「未知語」として検出される．これら未知語が文章を特徴付ける単語となっているケースはしばしば見受けられる．よって，blog 記事における特徴語抽出において，未知語を無視することはできない．そこで連続して「名詞」または「未知語」が出現する場合にはこれらを連結し，1 つの単語として扱う．

3.2.2 単語連結処理の副作用

複数の単語を結合して 1 単語とする場合の副作用として，単語の種類数が増加することがあげられる．たとえば「新潟中越地震」と「新潟県中越地震」という 2 種類の表現があったとき，計算機はこれらを異なる単語として認識してしまう．この問題への対処は従来から指摘されている表記ゆれや類義語の問題への対処と同様の困難さをともなう．しかし，単語の結合によりこのような副作用が発生する可能性はあるが，単語を結合しない場合よりは，意味が絞り込めるため，適切に関連性や相違性を求めることができると考えられる．

また，未知語を考慮することは適切な特徴語抽出のため重要であるが，新語や固有名詞だけでなく，形態素解析ミスによっても未知語と判定される単語もある．特に blog 記事は個人が書くため文章が文法的に正確でない場合が多く，形態素解析ミスも多くなる．これら 2 種類の未知語—“新語や固有名詞” と “形態素解析ミスによる未知語”—を計算機が区別することは困難であり，現状ではすべての未知語を対等に扱っている．その結果，単語連結処理において，意味の分からない単語が生成されてしまう可能性がある．現状ではこのような意味の分からない単語の除去はできず，この部分が本特徴語抽出における限界である．この問題への対処は今後の課題と考えている．

3.2.3 単語スコアに基づく特徴語抽出

単語連結処理後，文章中の単語の中から特徴語を選択するために，単語に単語スコアを付ける．記事 d に含まれる単語 w に対し tf-idf をベースとした以下の式により単語スコア $G(w,d)$ を定義する．

$$G(w,d) = \log \frac{tf(w,d)}{tf_{ave}(d)} \cdot \log \frac{1}{df(w)} \qquad (1)$$

ここで $tf(w,d)$ は記事 d における単語 w の頻度，$tf_{ave}(d)$ は記事 d における全単語の tf の平均値，$df(w)$ は全記事の中で単語 w の出現する記事頻度である．$G(w,d)$ がある閾値以上となる単語 w を特徴語とする．本研究では経験的に閾値を 8 とした．MineBlog において通常の tf-idf を使用しない理由は，$tf(w,d)$ の値は記事 d の文章の長さに依存するため，異なる長さの記事間での比較が行えないからである．そこで $tf(w,d)$ を $tf_{ave}(d)$ で割ることで正規化を行っている．また，tf は線形的に増えるので頻度によってはか

なり大きくなるため，対数をとることで高頻度となる単語の単語スコアを抑えている．

3.3 ノイズ記事の除去

3.3.1 推薦候補記事におけるノイズ記事

データベースから取り出した推薦候補記事の中には推薦記事としては不相応なノイズとなる記事が存在する．ノイズとなる記事としては，記事の抽出に失敗したページ，単語の羅列，to-do リストなどがあげられる．それぞれ次のような理由によりノイズ記事となる．

(1) **記事の抽出に失敗したページ**

MineBlog では blog ページから記事に相当する部分を自動的に抽出するが，この際 HTML の記述が著しく不正である場合には，HTML タグによってマークアップされている部分を正しく認識できず，複数の記事を 1 つの記事と見なし記事の抽出に失敗する．このため，記事抽出に失敗した記事を推薦結果として提示した場合，推薦には適さない記事も推薦されることになる．よって，記事抽出に失敗したページをノイズ記事と見なす．

(2) **単語の羅列，to-do リスト**

blog システムの利便性の向上により，blog を自分用のメモとして利用するケースがよく見られる．このようなことから，to-do リストが書かれた記事や，単語の羅列のみが書かれた記事が存在する．これらの記事の場合，本人にとっては重要な意味があるが，他人にとっては意味が分からないため推薦記事としては不適切である．このような記事はノイズ記事と見なし除去する必要性がある．

以上の 2 種類のノイズ記事を除去するために，単語数による手法と助詞の割合による手法を用い，いずれかの手法によりノイズ記事と判断したものを推薦候補記事から除去する．

以下 3.3.2 項で単語数によるノイズ記事の除去法，3.3.3 項で助詞の割合によるノイズ記事の除去法について述べる．

3.3.2 単語数によるノイズ記事の除去

単語数が多くなりノイズとなる記事としては記事の抽出に失敗したページがあげられる．そこで MineBlog では推薦候補記事 q の単語数 $m(q)$ を用いてノイズ記事を除去する．本稿では経験的に以下の条件を満たす推薦候補記事 q をノイズ記事と判断し除去する．

$$m(q) > 1{,}300 \tag{2}$$

単語数が多い記事を除去した場合，長文である記事も除去してしまう．長文である記事は新たな興味となる可能性はゼロではないが，1,300 語以上もの長文となると，多くのユーザにとって閲覧時に負担となる可能性のほうが高い．そのため長文記事を除去することは問題にならないと考える．

3.3.3 助詞の割合によるノイズ記事の除去

単語の羅列，to-do リストは記事の文章における助詞の割合が少なくなることから，助詞の割合を計算することでノイズ記事かどうかを判断し除去する．MineBlog では推薦候補記事 q の単語数 $m(q)$ に占める助詞の数 $p(q)$ の割合 $r(q) \equiv p(q)/m(q)$ を用いてノイズ記事かどうかを判断する．$r(q)$ が次の条件を満たす推薦候補記事 q を経験的にノイズ記事と判断し除去する．

$$r(q) < 0.12 \tag{3}$$

3.4 スコアリング

特徴語を抽出しノイズ記事を除去した推薦候補記事 q を，ユーザが書いた記事 d と比較しスコアリングする．このスコアリングにより上位となる記事を推薦記事と決定する．本節では推薦候補記事のスコアリングについて述べる．

3.4.1 スコアリング法

MineBlog では推薦候補記事に対し関連性，相違性，話題性の 3 つの尺度により評価し，推薦記事に含めるかどうかを判定する．3 つの尺度はそれぞれ新たに定義する関連度 $R(d,q)$，相違度 $D(d,q)$，話題度 $T(q)$ によって定量化され（3.4.2 項〜3.4.4 項），この 3 つの値によって各推薦候補記事をスコアリングする．ユーザが書いた記事 d と推薦候補記事 q に関するスコア $S(d,q)$ を次式で定義する．

$$S(d,q) = X \cdot R(d,q) + Y \cdot D(d,q) + Z \cdot T(q) \tag{4}$$

ここで X，Y，Z は定数である．

3.4.2 関 連 性

関連性とは，ユーザが書いた記事とどの程度関連する話題を含んでいるかを測る尺度である．記事に含まれる関連する話題がきっかけとなり，新たな興味につながると考えられるため，推薦候補記事に関連性がどの程度あるかを評価する．この関連性を定量化した量として推薦候補記事に対する関連度 $R(d,q)$ を次式で定義する．ただし，関数 G の定義は式 (1) のとおりである．

$$R(d,q) = \sum_i G(w_i^q, d) \tag{5}$$

ここで w_i^q は推薦候補記事 q に含まれる i 番目に高い単語スコアの特徴語である．$R(d,q)$ はユーザが書いた記事 d に含まれる特徴語を q がいくつ含んでいるかにより決定される．

3.4.3 相 違 性

相違性は，ユーザが書いた記事とどの程度異なる話題を含んでいるかを測る尺度である．関連性のみが高

い記事はユーザの書いた記事と似た内容のものとなるため，関連性のみによる評価では新たな興味につながる内容を含んでいるかどうかを測ることができない．そこで，推薦候補記事に相違性がどの程度含まれるかを評価する必要性が生じる．この相違性を定量化した量として推薦候補記事に対する相違度 $D(d,q)$ を次式で定義する．

$$D(d,q) = \sum_j \frac{G(w_j^q, q)}{N_q} \tag{6}$$

ここで関数 G の定義は式 (1) のとおりである．w_j^q は推薦候補記事 q において j 番目に高い単語スコアを持つユーザが書いた記事 d には含まれない特徴語，N_q は推薦候補記事 q における単語数である．$D(d,q)$ はユーザが書いた記事 d に含まれていない特徴語を q がどの程度の割合で含んでいるかにより決定される．

3.4.4 話 題 性

話題性とは，ウェブ上の blog 記事全体で一時的に頻繁に取り上げられた話題を記事中にどの程度含んでいるかを示す尺度である．ここでは，話題性を定量化した量として話題度を定義する．推薦候補記事 q に対する話題度を求めるために，blog に対応できるよう改良された burst 検出手法[5] を用いる．

burst 検出手法は，文章群をそれらに付加された時間情報（blog 記事であれば投稿日時）をもとに時系列に並べ，文章の間隔が時間的に狭くなっている時期を文章の頻出時期として検出する手法である．また，頻出時期において文章がどの程度頻出しているかを示す量は burst 度により定義される．この手法は単語の頻出時期の特定にも利用できる．具体的には，ある単語が含まれる文章のみを時系列に並べ，頻出時期を求め，それを単語の頻出時期とすることができる[5]．

そこで，推薦候補記事 q に対する話題度を求めるために，準備として，推薦候補記事 q が含む特徴語集合 $W(q)$ の中から，過去に頻出時期がなかった特徴語を除いた単語集合 $W'(q)$ を求めておく．この $W'(q)$ を利用し，推薦候補記事 q に対し話題度 $T(q)$ を次式により定義する．

$$T(q) = \sum_{w \in W'(q)} \big\{ B(w) \cdot N(w) \big\} \tag{7}$$

$$B(w) = F\big(O(w), -K_B, C_B\big) \tag{8}$$

$$N(w) = F\big(D(w), K_N, C_N\big) \tag{9}$$

$$F(x,K,C) = \frac{1}{1+e^{K(x-C)}} \tag{10}$$

ここで，式 (7)〜式 (10) における変数の意味は以下のとおりである．

- $B(w)$：単語 w の話題性の強さ
- $N(w)$：単語 w が話題になった時期の新しさ
- $O(w)$：単語 w の最も新しい頻出時期の burst 度
- K_B，C_B：定数，ただし K_B は正の定数
- $D(w)$：単語 w の最も新しい頻出時期とユーザが記事を投稿した時刻の差
- K_N，C_N：定数，ただし K_N は正の定数
- $F(x,K,C)$：入力変数 x，制御パラメータ K，C で定義されるシグモイド関数

$B(w)$ と $N(w)$ はともに式 (10) で定義される関数 F の出力であり，値域 $[0,1]$ をとる．関数 $F(x,K,C)$ は，シグモイド関数と呼ばれる単調増加型あるいは単調減少型の関数であり，入力変数 x と 2 つの制御パラメータ K，C によってその形状が決定される．x の増加あるいは減少にともない，関数の出力は 0 または 1 へと収束する．$x = C$ のとき関数の出力が 0.5 となる．また，K は $x = C$ における関数の傾きを表す．

式 (8) において，$O(w)$ はシグモイド関数における入力変数，K_B（正数），C_B はシグモイド関数における制御パラメータ（ただし K_B は符号を逆転して入力されるのでシグモイド関数は単調増加となる）となるので，$B(w)$ の値は $O(w)$ が大きいほど 1 に近付き，小さいほど 0 に近付く．

また，式 (9) において，$D(w)$ はシグモイド関数における入力変数，K_N（正数），C_N はシグモイド関数における制御パラメータ（K_N は符号を変えずそのまま入力されるのでシグモイド関数は単調減少となる）となるので，$N(w)$ の値は $D(w)$ が小さいほど 1 に近付き，大きいほど 0 に近付く．

以上まとめると，話題度 $T(q)$ は，推薦候補記事 q に含まれる全特徴語の中で，頻出時期における burst 度が高く，かつ，頻出時期が投稿日時に近い，という 2 つの性質を同時に満たす特徴語が多いほど大きな値をとる．

4. 評 価 実 験

本章ではシステムの有効性を評価するために行った実験について述べる．

4.1 実 験 方 法

普段から blog を書いている大学生および大学院生計 23 人を被験者として，1 人あたり 10 件の記事を書いてもらった．それに対し MineBlog は 1 件につき最大 10 件の推薦記事を被験者に提示した．なお，

MineBlog はデータベースに格納されている約 120 万記事の中から推薦記事を決定した．

複数の推薦方法を比較する目的で，実験では，データベースからランダムに取り出す方法（Random），関連度のみを用いる方法（R）—この方法は従来手法（文章をキーとする類似文章検索手法）と比較するという意図がある—，関連度と相違度を用いる方法（R+D），関連度と相違度と話題度（R+D+T）を用いる方法を使用した．推薦記事がどの方法で推薦した記事か被験者に分からないようにして提示した．

各被験者には 4 種類の推薦方法（Random，R，R+D，R+D+T）により推薦された各推薦記事を読んでもらい，その後，アンケートに答えてもらった．アンケートとしては，推薦された各記事を読んで新たな興味発見につながったかどうかを 5 段階（5：とてもつながった，4：ある程度つながった，3：どちらでもない，2：あまりつながらなかった，1：まったくつながらなかった）で回答してもらった．

被験者が記事を入力するために使用したインタフェースとして独自に改良した MovableType を利用した．従来の MovableType との違いは，ユーザが記事を投稿すると，それを MineBlog へと送り，さらに，MineBlog の出力である推薦記事を受け取る機能が追加されていること，ユーザが投稿した記事を表示するのと同時に MineBlog の推薦記事を表示できるようにしていること（2 章，図 2 参照）である．

なお，実験時の burst 検出における定数 s，γ [6] はそれぞれ 2，0.5 とし，MineBlog における定数 K_B，C_B，K_N，C_N（3.4.4 項，式 (8)，式 (9)）はそれぞれ経験的に 0.7，-3，0.4，20 とした．また，X，Y，Z（3.4.1 項，式 (4)）は 1.0，5.0，10 とした．X，Y，Z をこのように決定したのは，関連度，相違度，話題度が同程度の割合で評価されることを考慮した結果である．$R(d,q)$，$D(d,q)$，$T(q)$ の値域に差があることから，予備実験として 100 件の文章を使用し，$X \cdot R(d,q)$，$Y \cdot D(d,q)$，$Z \cdot T(q)$ の平均値が等しくなるように X，Y，Z を求めた結果，このような値の組合せになった．

4.2 実験結果

図 4 は，ランダムに推薦したもの（Random），関連度のみでスコアリングしたもの（R），関連度と相違度でスコアリングしたもの（R+D），関連度と相違度と話題度を用いてスコアリングしたもの（R+D+T）のそれぞれに対し，上位 3 位までに推薦した記事に対する被験者の評価をまとめたものである．

アンケートで 4 以上の評価を得た推薦記事—被験者が新たな興味につながったと判断した—の割合は，

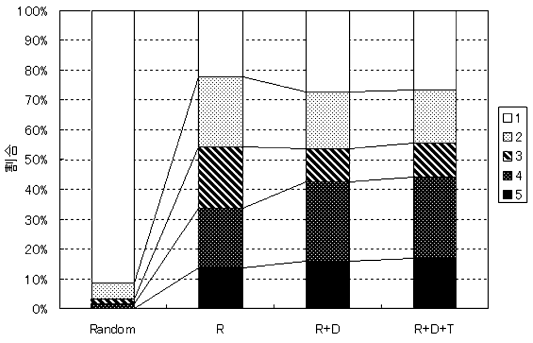


図 4 実験結果
Fig. 4 Results of experiment.

ランダムに推薦した場合は約 1.5%，関連度のみの場合は約 34%，関連度と相違度を用いた場合は約 43%，関連度と相違度と話題度を用いた場合は約 45%となった．この結果より，ランダムな推薦では新たな興味につながりにくいことは明確となった．関連度と相違度を用いることで関連度のみの場合と比較して高い推薦精度となり，関連度，相違度，話題度の 3 つすべてを用いることで最も良い推薦精度となることが確認できた．また，これは，提案手法を用いて推薦した記事のうち約 2 つに 1 つは新たな興味につながる推薦記事を提示できたことを示している．

4.3 実験結果の分析

4.3.1 関連度，相違度の有効性に対する分析

最初の分析として，関連度と相違度において，ユーザの主観に合った定量化が実現できているかどうかを確認するために，被験者に提示した各推薦記事に対して以下の質問に回答してもらった．

質問 1：自分が書いた記事と関連のある内容であったか．

質問 2：自分とは違う観点を持って書かれていたか．

この 2 つの質問に対し 7 段階（7：とてもそうである，6：ほぼそうである，5：だいたいそうである，4：どちらともいえない，3：あまりそうでない，2：ほとんどそうでない，1：まったくそうでない）で回答してもらい，被験者の評価値（1〜7）と関連度，相違度との相関性を調べた．

その結果，質問 1 の評価値と関連度の相関係数は 0.4043，質問 2 の評価値と相違度の相関係数は 0.4343 となり，正の相関性が見られた．これより関連度と相違度によって関連性，相違性が定量化できていることが分かる．よって，関連性，相違性という尺度で記事を評価できていることが分かる．

また，質問 1 において 5 以上の評価値が得られた推薦記事を対象に，質問 2 の評価値と，新たな興味につ

ながったかどうかに関する評価値（4.1 節参照）の相関性を調べたところ，相関係数 0.86 という高い相関性があることが分かった．このことから，関連性が高く相違性が高い記事は新たな興味につながることが分かる．

以上のことから，関連度と相違度を用いて新たな興味につながる推薦がなされていることが確認できる．

4.3.2 話題度の効果に関する分析

次の分析として，話題性による評価がどの程度推薦結果に影響を与えているかを調べた．図 4 の実験結果において，関連度，相違度，話題度の 3 つすべてを利用した推薦結果に対する被験者の評価と，関連度，相違度の 2 つを利用した推薦結果に対する被験者の評価を比較しても，新たな興味につながる記事の推薦精度は大差がないように見える．また，実際推薦されている記事を比較してもそれほど違いはなく，ほとんどの記事が関連度と相違度によって決定されている．しかし，話題度が高いため上位に推薦された記事に対する被験者の評価を調べると，約 62%の記事が新たな興味につながった記事（アンケートで 4 以上の評価）であったことが確認できた．このことから，話題度の高い記事は数は少ないものの，新たな興味につながりやすいという点において，話題度を考慮したスコアリングは意味があるといえる．

4.3.3 ノイズ記事の除去効果に関する分析

さらに，ノイズ記事の除去（3.3 節）効果を測るために，フィルタリングをした場合としない場合とでそれぞれ 300 件ずつの推薦記事を用意し，ノイズ記事の割合を調べた．その結果，フィルタリングをしなかった場合には約 34%のノイズ記事が含まれたのに対し，フィルタリングをした場合では約 8%となり，フィルタリングの精度は 77%となった．このことから，フィルタリングを行うことでノイズ記事を除去し，推薦精度の向上につながったといえる．

4.4 考　　察

4.4.1 関連性と相違性の有効性と問題点

実験結果より，関連性のみを用いて推薦記事を決定した場合より，関連性，相違性の 2 つを用いた場合のほうが精度の高い推薦ができている．関連性のみで推薦記事を決定する場合，ユーザの投稿した文章や推薦対象となる文章の性質によって検索結果は大きく異なる．たとえば，技術文章が推薦対象となった場合，ほぼ同じ内容の文章は少ないため，関連性のみを評価したとしても，異なる内容も含んでいる文章を推薦可能な場合が多いと考えられる．しかし，映画の感想について書かれた文章などが推薦対象となった場合，どれも似通った内容の文章ばかりとなり，異なる内容を含んだ文章が推薦されにくいと考えられる．特に blog 記事の場合，「おもしろかった」「楽しかった」などの簡単な感想を述べただけの短い文章が多く，どれも似通った文章となっている傾向がある．新たな興味の発見へとつなげるという観点からは，異なる内容が含まれていることが必要と考えられるので，そのような似通った文章が推薦されたとしても意味がない．実際，MineBlog で関連性のみを用いて，技術文章をもとに推薦した場合と，映画の感想の文章をもとに推薦した場合とで推薦結果を比較してみた．その結果，前者の推薦結果には異なる内容を含み新たな興味につながる記事が多かったが，後者の推薦結果には異なる話題に触れるような記事（たとえば，別の映画との比較をしているような記事）は少なかった．以上のことから，関連性のみを用いた場合，簡単な感想を述べた記事をもとにした推薦において，推薦精度が低くなったと考えられる．

一方，MineBlog では明示的に相違性を用いて異なる話題が含まれているかどうかを評価することができるので，同じものに対する感想のみを述べた記事が推薦結果から除去され，推薦精度が向上すると考えられる．

しかし，実験結果から，アンケートにおいて 4 以上の評価を得た記事が増えるとともに，2 以下の評価を得た推薦記事も若干増えていることが分かる．MineBlog では，ユーザが新規に投稿した記事だけしか見ておらず，過去に投稿した記事の情報を利用していない．その結果，MineBlog は，ユーザが過去に投稿していた記事に含まれる話題と同じ話題が含まれた記事を推薦してしまうことがある．相違性が高いと評価された話題であっても，ユーザが過去に書いた話題と同じような内容の場合，新たな興味につながる記事とはならない．そのため，相違度を計算する際，過去にユーザが書いた記事や，過去に推薦した記事も調べ，ユーザが既知の話題を評価しないように改善する必要があると考えられる．

4.4.2 話題度の問題点

次に話題度が相対的に高くなる記事が少ない原因について考察する．話題度を算出するためのもとになっている burst 検出手法では，ある期間に新規に追加された blog 記事において，単語の出現割合を調べ，この値が過去の期間に比べ急増しているかどうかを判定し，一時的に頻繁に使われた単語を特定する．しかし，たとえば大事件の後などで，ある期間で新規に追加される blog 記事数が過去の期間の blog 記事数と比較

し急増する場合がある．このような期間においては，急増した記事のほとんどがその大事件について触れており，それ以外の話題の割合は相対的に低くなってしまう．そのため，それ以外の話題に含まれる単語に出現頻度が増加した単語があっても，その単語の出現割合は増加しないため burst として検出されない．このため，多くの話題を扱うことができなかったと考えられる．一定期間に存在する全 blog 記事における単語の出現割合ではなく，特定の趣味嗜好のユーザが投稿した記事における割合を算出するなどして，より高い精度で話題性を計測可能にする必要性があると考えられる．

5. 関連研究

blog を対象とした研究として blogWatcher[7] がある．blogWatcher は blog を情報源として注目し，キーワードによる blog 検索，blog からのホットなキーワードの抽出，blog からの評判情報の抽出を行うことができる．MineBlog とは違い，新たな興味を喚起する記事を積極的に推薦することはしない．

blog 記事を自動的に推薦するサービスとして Bulkfeeds☆，News & Blog Search ☆☆，So-net blog ☆☆☆などがある．これらのサービスは MineBlog と同様に，ユーザが書いた記事を用いて推薦する blog 記事を決定する．しかし，いずれのサービスも似ている内容の記事を探し出しそれをユーザに推薦するというものであり，MineBlog で用いたような相違性，話題性といった尺度を用いた推薦記事の決定は行っていない．

また，ウェブページを推薦するシステムとしてウェブナビゲータがある[8]．このシステムでは，ユーザの閲覧履歴を利用することでユーザの嗜好を学習し，その嗜好に適したものをあらかじめ用意されたウェブページの中から選択し，ユーザに推薦する．このシステムはあらかじめ用意された 500 のページの中からのみ推薦するページを決定するため，ウェブをクローリングするといったことは行っていない．

6. おわりに

本稿では，ユーザが書いた記事をもとに新たな興味発見につながる blog 記事を推薦するシステム MineBlog について述べた．blog 記事を評価する尺度として，関連性，相違性，話題性を提案し，それぞれ新たに関連度，相違度，話題度を定義し定量化を行った．また，評価実験から，これらの尺度を用いたスコアリングによって blog 記事を推薦することで，推薦する記事のうち約 2 つに 1 つの記事は新たな興味につながることを示した．

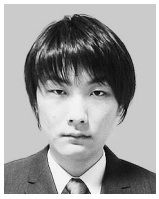
今後は，過去に書かれたユーザの記事に含まれるユーザの興味や嗜好を利用したり，推薦記事に対する良し悪しのフィードバックなどを利用したりすることで推薦のパーソナライズ化を行い，推薦精度の向上を図りたいと考えている．

☆ Bulkfeeds, http://bulkfeeds.net/
☆☆ News & Blog Search, http://news.drecom.jp/
☆☆☆ So-net blog, http://blog.so-net.ne.jp/

参考文献

1) Lindahl, C. and Blount, E.: Weblogs: Simplifying Web Publishing, *IEEE Computer*, Vol.36, Issue 11, pp.114–116 (2003).
2) Herring, S.C., Scheidt, L.A., Bonus, S. and Wright, E.: Bridging the Gap: A Genre Analysis of Weblogs, *Proc. 37th Hawaii International Conference on System Sciences*, p.40101b (2004).
3) Nardi, B.A., Schiano, D.J., Gumberecht, M. and Swarts, L.: Why We Blog, *Comm. ACM*, Vol.47, No.12, pp.41–46 (2004).
4) 西尾章治郎ほか：情報の構造化と検索，p.115，岩波書店 (2000).
5) 藤木稔明，南野朋之，鈴木泰裕，奥村 学：document stream における burst の発見，情報処理学会研究報告，2004-NL-160，pp.85–92 (2004).
6) Kleinberg, J.: Bursty and hierarchical structure in streams, *Proc. 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.91–101 (2002).
7) Nakano, T., Fujiki, T., Suzuki, Y. and Okumura, M.: Automatically Collecting, Monitoring, and Mining Japanese Weblogs, *Proc. WWW2004 Conference*, pp.320–321 (2004).
8) 九津見洋，内藤榮一，荒木昭一，江村里志：ユーザ適応型ホームページ推薦ソフトウェブナビゲータの開発，信学会論文誌 D-II，Vol.J84-D-II, No.6, pp.1149–1157 (2001).

(平成 17 年 4 月 1 日受付)
(平成 18 年 1 月 6 日採録)



森本　和伸

1980 年生．2003 年電気通信大学電気通信学部情報工学科卒業．2005 年同大学大学院電気通信学研究科情報工学専攻博士前期課程修了．同年富士通株式会社入社，現在に至る．在学中は情報検索の研究に従事．

林　　貴宏（正会員）

1975 年生．1998 年金沢大学工学部電気情報工学科卒業．2000 年同大学大学院自然科学研究科博士前期課程修了．2003 年同研究科博士後期課程修了．博士（工学）．2001 年石川工業高等専門学校電子情報工学科助手を経て，2003 年電気通信大学電気通信学部情報工学科助手，現在に至る．マルチメディアデータベース，情報検索，人工知能の研究に従事．IEEE，電子情報通信学会，日本ソフトウェア科学会，人工知能学会各会員．

尾内理紀夫（正会員）

1950 年生．1973 年東京大学理学部物理学科卒業．1975 年同大学理学系大学院物理学専攻修士課程修了．同年日本電信電話公社（現 NTT）に入社．1982 年から 85 年に ICOT プロジェクトに参画，1997 年から 98 年に RWC プロジェクトに参画．2000 年より電気通信大学電気通信学部情報工学科教授．著書に『コンピュータの仕組み』（朝倉書店），編書に『オブジェクト指向コンピューティング III』（近代科学社），『インタラクティブシステムとソフトウェア V』（近代科学社）等．マルチメディア情報処理，情報検索，セマンティックコンピューティング等に興味を持つ．1996 年情報処理学会プログラミングシンポジウム山内奨励賞受賞．工学博士（東京大学）．日本ソフトウェア科学会，人工知能学会，ACM 各会員．