Blog における RSS リコメンデーションシステムの実現

2002MT003 安藤 隆成　　2002MT022 伊藤 優
指導教員　河野 浩之

1　はじめに

大量の Blog 記事の中から自分と同じ興味関心をもつものを探すことは容易ではない．これらの情報を探すためには，検索エンジンを用いる方法か，トラックバック先の記事を辿る方法が考えられる．しかし，どちらの方法にも問題点がある．前者は膨大なページの中から探す手間がかかる．また，検索エンジンではインデックスの更新に時間がかかり速報性のある Blog 記事にとって有効な検索手段とはいえない．後者では，自分の記事を参照しトラックバックをしてもらえない限り自分と同じ様な嗜好を持った記事を探すことはできない．

そこで本研究では，これらの問題点に注目し，Blog における類似記事のリコメンデーションシステムを実装する．

2　現在の Web マイニング技術と Blog 研究

Blog 記事の単語の burst を求めることにより，その日に注目されている話題や Blog を提示する blogWatcher::meta-blog が奥村らによって開発されている [1]．meta-blog では，burst 度を全単語について計算することでその日に burst している単語のリストを得る．この単語を元に注目されている話題を発見し，その話題を Movable Type で作られた Blog に対して毎日記事を投稿している．

また，大薮らは，Web コンテンツマイニングによりページ間の類似性やカテゴリ分けが可能となるツールを開発している [2]．このシステムでは，各ページの単語の重要度によるドキュメント間の類似性の検証を行うことにより，類似する Web コミュニティの発見とカテゴリ分けを行っている．

3　RSS リコメンデーションシステム

3.1　リコメンデーションの流れ

本システムはユーザの Web サイトに組み込まれたシステムを前提としているので，まずサイト環境を構築しそこにリコメンデーションシステムを実装する．サイトは，Apache, Movable Type[*1] を用いて構築する．また，データベースに MySQL を使用する．システム全体の流れを図 1 に示す．

図中の (3) は，収集した RSS Feed を 1 つの記事単位に分割したデータが格納される．これと投稿記事のそれぞれに対し形態素解析と前処理を行いデータベースに格納する (6)(7)．(6)(7) を比較・分類し，その結果 (9) から類似する記事の情報を表示できるようにする．

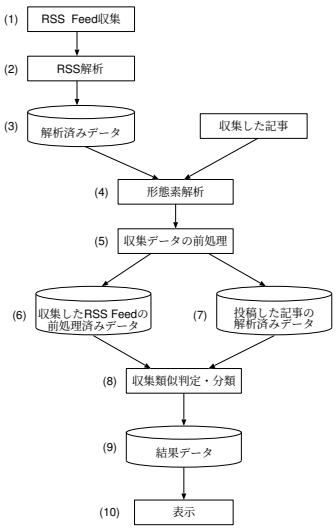


図 1　リコメンデーションの流れ

3.2　RSS Feed の収集

図 1 の (1) にあたる処理である．本研究では Web 上から収集する情報は RSS1.0(RDF Site Summary) とする．RSS Feed には図 2 のように，そのサイトの記事のタイトル，投稿日時，記事の概要や最近投稿した複数の記事情報などが記述されており，ネットワークへの負荷を少なくし一度に多くの情報を収集することができる．

RSS Feed の収集は毎日多くの記事の更新が通知される ping サーバに対して行う．本システムでは，ping サーバから RSS が公開されているページを取得する．そこから各 Blog の RSS Feed が置いてある URL を抽出し，RSS Feed を取得する．

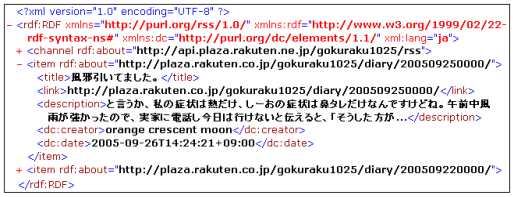


図 2　収集された RSS Feed の例

3.3　RSS Feed の解析

RSS 解析プログラムは収集された RSS Feed から記事ごとにタイトル (title)，更新日時 (dc:date)，概要

[*1] Six Apart-Movable Type, http://www.sixapart.jp/movabletype/

表 1　RSS Feed から抽出したデータの例

収集記事 ID	34
title	風邪引いてました。
link	http://plaza.rakuten.co.jp/gokuraku1025/diary/200509250000/
description	と言うか、私の症状は熱だけ、しーおの症状は鼻タレだけなんですけどね。午前中風雨が強かったので、実家に電話し今日は行けないと伝えると、「そうした方が…
date	2005-09-26

(description)，その記事の URL(link) を抽出しデータベースに格納される (表 1)．収集記事 ID とは，収集記事に対して収集順に一意に付けられた番号である．

3.4　形態素解析

RSS Feed 解析されたデータとユーザが投稿した記事に対して形態素解析を行う (図 1(4))．解析対象は記事のタイトルと概要とする．これには茶筌[*2]を用いる．前処理では名詞，自立動詞，自立形容詞を抽出する．(図 1(5))．

記事の解析，前処理が済むと記事毎にテーブルを作成し単語のみを格納していく．また，前処理後に残った単語は，単語 ID という一意の番号を振り単語リストテーブル (単語 ID，単語，DF を一組とするテーブル) へ格納する．また，DF(Document Frequency) を求めテーブルへ格納する．このテーブルは類似度計算のために使うテーブルである．類似度の計算方法は次節で説明する．

3.5　類似度の計算

前処理済みデータからのデータ分類には一般的な tfidf 法を用いて記事の類似判定を行う．

tfidf 法の TF(Term Frequency) と IDF(Inverted Document Frequency) は次式のようになる．文書 d 中の単語 t における tf 値は，

$$tf(d,t) = log(\frac{f_d^t}{f_d} + 1) \tag{1}$$

f_d^t は文書 d における単語 t の出現回数，f_d は d 中に含まれる全ての単語の出現回数の総和を表している．単語 t の idf 値は，

$$idf(t) = log(\frac{N}{N_t}) \tag{2}$$

N は投稿記事, 収集記事を合わせた全記事数, N_t は単語 t が出現した記事数を表し，DF(Document Frequency) ともいう．この値は単語リストテーブルの df に格納されている．上式 (1)(2) より，$tfidf$ 値は

$$tfidf(d,t) = tf(d,t) * idf(t) \tag{3}$$

よって文書ベクトル d_i は以下のようになる．

$$\vec{d_i} = \{ftidf_{1k}, ftidf_{2k}, \cdots, ftidf_{ik}\} \tag{4}$$

k は全文書に含まれる単語数，$ftidf_{ij}$ は文書 D_i 中の単語 t_j の $tfidf$ 値である．

[*2] ChaSen's Wiki, http://chasen.naist.jp/hiki/Chasen/

類似度は式 4 で求められた 2 つの文書ベクトル d_i, d_j の内積によって求める．すなわち，類似度 $simi$ は，

$$simi = \frac{\vec{d_i} \cdot \vec{d_j}}{|\vec{d_i}| * |\vec{d_j}|} \tag{5}$$

となり，2 つのベクトル d_i, d_j はそれぞれユーザが投稿した記事と収集した記事情報である．

投稿した記事の前処理済みデータと収集記事の前処理済みデータから類似度 (式 5) を求め，分類テーブルに格納していく．

3.6　結果表示

推薦記事は分類テーブルを用いて Movable Type 上に掲示される (図 1(10))．リコメンデーションする記事の表示は，類似度の高い順に表示させるようにする．類似度が同じであった場合は新しく収集された記事を上位とする．結果表示は Movable Type のプラグインとして作成・登録し，投稿記事毎のページへ出力するようにした．

4　実装したシステムの評価と考察

4.1　実装結果

実装したシステムを示す．図 3 は Blog のトップページの一部である．ここには複数個の記事が掲載されている．一つの記事の下側 (トラックバック，コメントの欄) に，推薦記事という文字と数字がある．ここをクリックすると，図 4 のような記事の個別ページへ飛び，投稿記事の真下に複数の推薦記事が閲覧できる．

4.2　評価の目的

本システムに対する評価は，アンケートにより推薦記事の似ている度合いを評価してもらい，本システムの有効性を検証する．また投稿記事ごとの類似度による推薦記事数から妥当な閾値を考察する．

4.3　評価 1：アンケート評価

実装した Blog の投稿記事は全部で 40 記事がある．記事の内容は，投稿者自身の身の回りに関した事柄について書かれた記事とニュースやイベントなど投稿者自身に直接関わりのない事柄について書かれた記事の 2 種類に分けることとする．これと収集した 68,843 個の RSS Feed の記事情報を類似判定し，各投稿記事との類似度が高い上位 10 記事を推薦表示させた．研究室の学生 14 人に協力してもらい，投稿記事と推薦記事の内容に類似性があるかどうかをアンケートによって評価した．評価は 1〜9 の 9 段階で，数値が高いほどよく似ているとする．評価してもらう記事は，40 記事中特徴のある内容であった 2 種類の記事それぞれ 5 つの計 10 記事である．以下に評価結果の散布図，相関係数 r の一例を示す．

4.3.1　投稿者自身の身の回りに関した事柄について書かれた記事

5 記事ともバラツキがあり評価の平均が全体的に低い値を取った．どれも投稿記事と同じ単語が含まれてはい

紅白歌合戦

おざっきーです。学校にきたので、書き込みしていきます。
昨日のライヴたのしかったです。
明日はELLEGARDENのライヴ行ってきます。
紅白といえば和田アキ子が白組でゴリエが紅組ですね。

限界です(笑)
まったく興味ありません。

投稿者: g302lab 日時: 13:45 | パーマリンク | コメント (0) | トラックバック (0) | 推薦記事 (10)

2005年12月05日

ライヴ漬け

おざっきーです。

図 3　Blog トップページの一部

紅白歌合戦

おざっきーです。学校にきたので、書き込みしていきます。
昨日のライヴたのしかったです。
明日はELLEGARDENのライヴ行ってきます。
紅白といえば和田アキ子が白組でゴリエが紅組ですね。

限界です(笑)
まったく興味ありません。

投稿者: g302lab 日時: 2005年12月06日 13:45 | パーマリンク

推薦記事

アキコ。

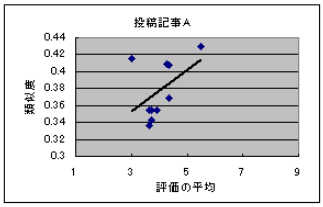
紅白歌合戦で和田アキ子。男性チームで登場。和田アキ子が紅白に出場する事は否定しないよ。和田アキ子紅白歌合戦みたいな大物加減あるからさ。だが、しかしですよ。男チームで登場って。待てと。確かに、和田アキコは男か女かどっちつかずさんですよ。ハー…

URL: http://blog.livedoor.jp/pinkostar/archives/50250775.html

2005-12-08T17:59:39+09:00

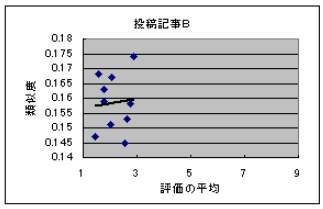
図 4　推薦記事の表示

るが，内容の異なった記事が多く推薦される結果となった．以下に 2 例を示す．投稿記事 A は投稿記事に含まれていた “ライヴ” という単語が推薦記事全てに入っていたため，類似度は比較的高めの値をとった．しかし，類似度が高くても人の評価は低くなった部分もあった．これは，特徴的な単語により類似度が高くなっても，内容からは似ていると判断し難いと判断されたと考えられる．投稿記事 B は類似度も人の評価も低くなったが，ほとんど関連が見られなかった．この記事もライヴという単語が含まれていたが，推薦記事にはほとんど含まれておらず “バンド” など他の単語が入っているものが多かった．こうなった原因は，投稿記事 B よりも A との類似度が高いものが多かったため “ライヴ” が特徴的な単語となった記事は A に推薦されてしまったと考えられる．また，“バンド” という語でも同音異義語でるアクセサリーのバンドが話題の記事が推薦されたことも評価が低くなった原因の一つと言える．



r=0.475

図 5　投稿記事 A の評価結果



r=0.0837

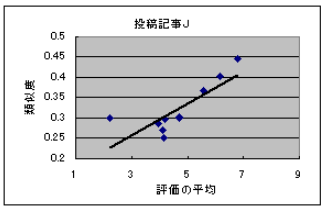
図 6　投稿記事 B の評価結果

4.3.2　投稿者自身に直接関わりのない事柄について書かれた記事

5 記事のうち 3 記事は投稿記事 C，残りの 2 記事は投稿記事 D のような結果となった．以下に 2 例を示す．投稿記事 C はニュースで話題になったもの，一時的に流行となった話題について書かれていたので，関連性のある記事が推薦されいた．また，評価の平均の幅が広い．これは，同じ単語が多く含まれていたが，内容は似ていると言えないものもあったため，そのような記事には低い評価が付けられたからだと考えられる．投稿記事 D では残差が多くバラツキも大きいので，相関係数の値は高めに出ているものの相関はほとんどないといえる．記事も評価の平均に幅があり，両端に極端に分かれた．これも先に述べた原因が強く表れたのではないかと考えられる．

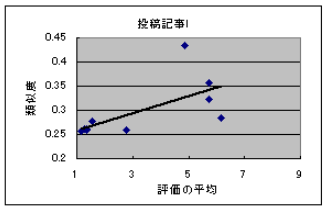
4.3.3　まとめ

以上の結果から，本システムの推薦はニュースやイベントなど投稿者自身に直接関わりのない事柄について書かれた記事にはある程度有用性が認められた．しかし，投稿者自身の身の回りに関した事柄について書かれた記事は，投稿者の趣味に関する内容が含まれいることが多く，これらの記事に類似性の高い記事を推薦しなければならないはずであるが，有用な推薦がされなかった．また，文書長が短過ぎてほとんど似ていない記事やその文だけでは記事の中身が掴めないものが推薦 (上位に位置) されるものが多く含まれていた．これは，RSS Feed の description は本文の先頭から数十文字を引用しているものが多く，日本語記事では文章の先頭にトピックセンテンスがこないこと，文字列長が短いものが多く一つの単語のスコアが高くなってしまったことにより，投稿記



r=0.806

図 7　投稿記事 C の評価結果



r=0.656

図 8　投稿記事 D の評価結果

事のスコアの低い単語と収集記事のスコアの高いそれによって類似度が決まってしまい，ほとんど似ていない記事が推薦されるということが生じたのではないかと考えられる．また，ニュース記事以外のほとんどの記事は口語表現で書かれており，うまく形態素解析ができなかったことも原因であるといえる．

システムの精度を上げるには，現在の RSS Feed の description での推薦は困難で Feed のリソースを辿って記事そのものを解析するか，抜粋でなく記事の要約が記述できるような新たな site summary が規格されるのを待つしかない．口語表現がうまく解析できる辞書やシソーラス, 別の類似度計算方法を用いることが上げられ, 今後の課題とされる．

4.4　評価 2：類似判定の閾値

図 9 は類似度別の推薦記事数をグラフにしたものである．評価 1 の結果では，類似度 0.35 あたりから評価の平均が比較的高くなっており，類似度の閾値を 0.35 にすれば本システムでの類似性が高いと見られる記事の推薦が可能となる．しかし，類似度 0.35 以上では推薦記事が存在する投稿記事の数が少なくなってしまう．逆に閾値を 0.35 より小さくすると，図 9 からわかるように推薦記事数が多くなるが，類似性が見られない記事も多く含まれるため，システムの精度を下げてしまうことになる．

システムの精度を下げずにより多くの記事を推薦するためには，全ての投稿記事に同じ閾値を当てはめるのではなく，記事ごと，または記事の種類ごとに異なる閾値を用いる方法が考えられ，今後の課題となる．

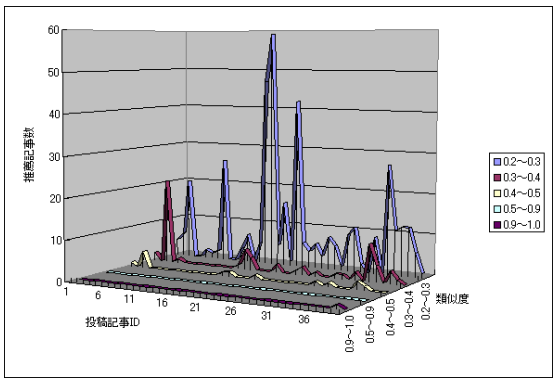


図 9　類似度別推薦記事数

5　おわりに

本研究では，ユーザが投稿した記事に類似する記事のリコメンデーションシステムを実装した．本システムが推薦する記事が，実際に似ているのかアンケートを行ってシステムを評価した．その結果，投稿者自身に直接関わりのない事柄について書かれた記事にはある程度有効であった．特に，RSS Feed の記事情報のみで類似判定を行うことが有効でないこと，Blog 記事は口語表現であるため形態素解析に用いる辞書を工夫しなければならないということが判明した．閾値の調査では，投稿される記事には内容にいくつが特徴があり，全ての類似判定を一つの閾値で行うことが困難であることが分かった．さらに，類似度が 3 割から 4 割という少ない値で人はある程度似ていると判断しており，類似度の数値と人の感覚に差があることが判明した．今後の課題として，推薦記事の収集・本文の抽出方法の考案や形態素解析辞書の改良，類似度計算の工夫が上げられ，さらなる研究が必要である．

謝辞

本研究を進めるにあたり，御指導頂いた河野浩之教授，貴重な時間を割いて推薦記事評価のアンケートに御協力して頂いた研究室の学生に深く感謝致します．

参考文献

[1] 奥村学, 南野朋之, 藤木稔明, 鈴木泰裕 “blog ページの自動収集と監視に基づくテキストマイニング,” 第 6 回セマンティックとオントロジー研究会資料, A401-01, pp.1-8, 2004.

[2] 大薮永, 小柳滋, “Web コンテンツマイニングによるページ間の類似性の判定ツール,” 第 3 回情報科学技術フォーラム, D-046, pp.107-108, 2004.