blogとニュース記事の自動対応付け

池田 大介† 藤木 稔明†† 奥村学‡
†東京工業大学 工学部情報工学科 ††東京工業大学 大学院総合理工学研究科
‡東京工業大学 精密工学研究所
{ikeda, fujiki}@lr.pi.titech.ac.jp oku@pi.titech.ac.jp

1 はじめに

近年，blog(Weblog) が急速に普及してきており，情報源としての期待が高まってきている．blog は個人が簡単に情報を発信することが可能なツールである．そのため，blog では書き手の興味のあることや関心のあることに対する意見や感想が述べられることが多い．

blog ではニュースに対する意見や感想が述べられることもよくある．しかしそういった場合，普通 blog にはニュースの詳細は書かれない．読み手が言及されたニュースについて詳しく知らなかった場合，blog の内容を理解するにはそのニュースについて詳しく書かれた記事を読む必要がある．近年，新聞社の Web サイト等から簡単にニュース記事を入手することができるようになってきたが，逆に情報が多すぎることもあり目当ての記事を簡単に入手できるとは限らない．ニュースについて言及している blog に，ニュース記事へのリンクが張られていれば読み手はリンクを辿ることで簡単に目的のニュース記事を入手できるが，リンクが常に用意されているとは限らない．リンクが用意されていない場合に，それを自動的に補うことができれば，この問題を解決することができる．

また逆に，自分が注目しているニュース，関心のあるニュースに対し，どのような意見があるかが知りたいという要求もある．この場合，あるニュースに対して，それについて言及している blog を求めることができれば，それらを見ることで多くの人の意見や感想を収集することが可能である．blog 検索エンジン等を使うことでニュースについて言及された blog を見つけることも可能だが，blog は急速に増えており，求める blog だけを漏れなく収集することは難しくなってきている．

これらの要求は，ニュースについて言及された blog と，そのニュース記事の間の対応付けを行うことで同時に実現可能である．すなわち，blog からそれに対応づけられているニュース記事を見ることでリンクを自動的に張ることができ，ニュース記事から対応づけられた blog を見ることで注目しているニュースについて言及された blog を過不足なく収集できる．

文書間の対応付けは類似度を用いることで実現できる．文書間の類似度の計算法として，Cosine 類似度が良く知られている．しかし blog とニュース記事のような，性質の大きく異なる文書間にはそのままでは適用できない．そこで本研究では，両者の性質の違いを考慮することによって，対応付けの性能を大きく改善できる手法を示す．

2 関連研究

Itoh ら [2] は，blog とニュース記事のような異なる種類のデータベース間の検索技術として，Term Distillation を提案している．異なる種類のデータベース間では，クエリとなるデータベースと，検索対象となるデータベース中の語の出現頻度の分布が異なる．そのため，本来あまり重要でない語に対して大きな重みがついてしまうことがある．Term Distillation とはこのような事態を防ぐため，望ましくない重みがついてしまう語を始めから検索に用いないようフィルタリングする技術である．

具体的には，検索側の語をクエリ候補とし，各クエリ候補 t に対し，TDV_t を求める．TDV_t は以下の式で与えられる．

$$TDV_t = tf \cdot \frac{p(1-q)}{q(1-p)} \tag{1}$$

ただし p は検索対象データベース中の語 t の出現確率，q はクエリデータベース中の語 t の出現確率，tf はクエリ中の語 t の出現回数である．クエリ候補のうち，この TDV の高いものから一定の数を検索に用いる．

Itoh らは，クエリデータベースとしてニュース記事を，検索対象データベースとして特許データベースを用いた実験に Term Distillation を適用し，成果を挙げている．

3 提案手法

3.1 システムの概要

blog とニュース記事の対応付けは，大きく以下の 3 ステップによって成される．

1. ニュース記事からの特徴語ベクトルの生成
2. blog ベクトルの生成
3. ニュースと blog のベクトル間の類似度の計算

まずニュース記事，blog の両方から語ベクトルを生成する．それらのベクトル間の類似度を計算し，類似度

が閾値以上のニュース記事と blog の組を，その blog はニュースについて言及しているとみなし，対応付けする．

以下ではそれぞれのステップについて詳しく説明する．また，3.5 節ではニュースの特徴語ベクトルを生成する際に，語の出現頻度の推移を用いて重み付けする手法を説明する．

3.2 ニュース記事からの特徴語ベクトルの生成

blog とニュース記事の対応付けを行う場合，書かれる内容の性質の違いが問題となる．ニュース記事ではニュースやイベントについて詳細に記述されるのに対し，blog ではニュースやイベントについては，それがどのニュースか特定できる程度の情報しか書かれず，書き手の意見や感想などが文書の主となることが多い．そのため，ニュース記事とそれに言及している blog の間で共通して現れる語はそのニュースを特定することができるような語だけである．ニュース記事から特徴語ベクトルを生成する際には，そのような語だけを選ぶ必要がある．

ニュース記事は通常，タイトルでニュースの全体像がつかめ，最初の 1 文が内容のサマリとなっており，具体的な内容まで分かるよう書かれている．2 文目以降にはニュースの背景や詳細などが書かれることが多い．したがって，前述のようなニュースを特定できる語は通常，タイトルと最初の 1 文目に集中しており，2 文目以降にはあまり blog には使われない語が多いと考えられる．そのため，特徴語ベクトルの語の候補として，タイトルと 1 文目に含まれる語を使用する．

語の重み付けは TFIDF 法がよく知られている．本研究でもこれを用いた語の重み付けを基本とした．重み付けされた特徴語ベクトルのうち，重みの上位一定数を特徴語ベクトルとする．

3.3 blog ベクトルの生成

先述した通り，blog ではニュースについて言及する際ニュースについては詳しく述べないことが多い．ニュースについて言及していても，そのニュース記事中に現れる語が繰り返し何度も使われるとは限らない．そのため，blog 中の語の出現回数はあまり意味をもたないと考えられる．よって，blog ベクトルとしては IDF だけで重み付けされたベクトルを用いる．blog 中の語のうち，どれがニュースを記述する際に使われるかを特定することはできないので blog ベクトルには blog 中の語が全て含まれる．

3.4 ベクトル間の類似度の計算

生成された二つのベクトルから類似度を求める手法として Cosine がよく使われる．Cosine はベクトルの内積をベクトルの長さで正規化することで文書の長さに依らない類似度を求めることができる．

しかしこれは，類似する文書ならば長い文書は短い文書に比べて内積が大きくなるという仮定に基づいている．blog では一つの記事の中で複数の事柄について述べていることが少なくない．こういった場合, 記事自体が長いにも関わらず, ニュースについて言及している部分は少ない，ということがある．そのため, Cosine のような文書の長さで正規化するような類似度は正しく対応付けできない場合がある．したがって，blog とニュース記事の対応付けの場合は，正規化をしない内積が最も効果的であると考えられる．

3.5 語の出現頻度の推移による重み付け

ニュース記事と blog の対応付けをする際，ニュース記事の特徴語ベクトルとして重要なのは，blog でそのニュースについて記述される時によく使われ，他の場合にはあまり使われないような語である．このような語は blog に出現した場合，そのニュースについて記述している可能性が高い．したがって，このような語に大きな重みを与えることが，ニュース記事の特徴語ベクトルを生成する際に必要である．

しかし，TFIDF はそのニュースが記述される際に使われやすい語に対し，必ずしも大きな値を与えるわけではない．TFIDF によってニュース記事中の特徴的な語を抽出することが可能だが，それは他のニュース記事に比べて特徴的な語であるというだけで，そのニュースについて記述する際に必要な語とは限らない．もちろん多くの場合，ニュースにおける特徴的な語と，記述する際に使われる語とは合致する．しかしそうでない語があると，本来重要であるはずの語の重みが小さくなったり，その逆が起こったりする．これは対応付けのミスの大きな原因となり得る．

あるニュースが blog で言及される場合，その多くはニュースの直後に書かれる．つまり，あるニュースについて言及する際に使われやすい語は，そのニュースが書かれた直後に出現頻度が高まると考えられる．先述したように，このような語には特別な重みを与えることで良い結果が期待できる．

言及するニュースが有名なものであればあるほど，blog ではそのニュースについての詳細が記述されなくなる傾向がある．有名なニュースであればその詳細まで書かなくとも，読み手はそれがどのニュースについて書かれているのか理解できると考えられ，この傾向は自然である．

このとき，通常の TFIDF のみの重み付けだと，使用される語が少ないため総じて低いスコアになってしまう．出現頻度の高まった語に特別な重みを与えることで，有名なニュースに使用される語の重みは特に大きくなり，この問題を解消できると考えられる．

例を示す．11 月 1 日，新札の流通が始まり，ニュース記事ではもちろん，blog でもよく取り上げられた．新札は肖像の人物が変更されたため，ニュース記事ではそれが取り上げられ，“札” や “肖像” が特徴的な語として選ばれる．

それに対し，blog では単に “お札が新しくなった” とだけ述べて肖像については言及しなかったり，言及していても “肖像” という語は使われなかったりする．

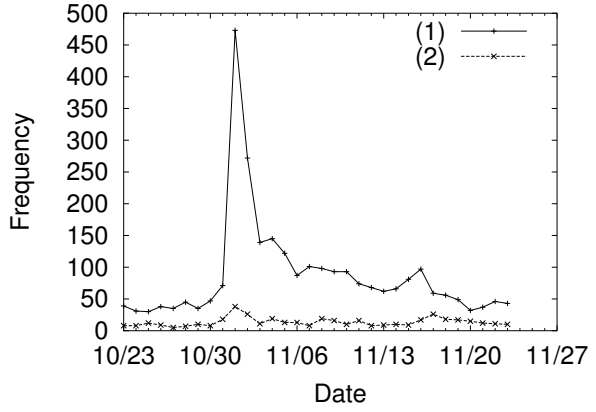

図 1: blog 中の (1)“札” と (2)“肖像” の出現頻度の推移

図 1 は blog 中の “札” と “肖像” の出現頻度の推移を表したグラフである．“札” が新札の流通が始まった 11 月 1 日から数日使用頻度が大きく増加しているのに対し，“肖像” は 11 月 1 日付近で増加こそしているものの，“札” と比べて変化量が少なく，blog でこのニュースについて言及される際にあまり使われていないことが分かる．したがって，このニュース記事から特徴語ベクトルを生成する場合，“札” に大きな重みをつける必要があると考えられる．

具体的には，ニュース記事中の語 t に関して，そのニュースが書かれた直後の IDF が他の期間の IDF と比べて特に減少している場合，語 t はそのニュースについて blog 中で記述された結果，出現頻度が高まり IDF が減少したと考えられる．したがって IDF の減少分を考慮し，重みに反映させる．

ある期間 d の語 t の IDF を $IDF(t,d)$ として,

$$IDF_{sub}(t,d) \quad = \quad IDF(t) - IDF(t,d) \qquad (2)$$

とする．d をニュースが書かれた日から 3 日間とすることで, そのニュースが登場した後の語の出現頻度の変化を知ることができる．

この IDF_{sub} をニュース記事特徴語ベクトル中の各単語についてに求め，それと TFIDF による重みの和を，その単語の重みとする．つまり，ニュース記事ベクトル中の語 t の重み $w(t)$ は，語 t の TFIDF による重みを $TFIDF(t)$ として,

$$w(t) \quad = \quad IDF_{sub}(t,d) + TFIDF(t) \qquad (3)$$

と表せる．

4 実験

4.1 実験方法

20 件のニュースをランダムに選び，各々について対応付けされる blog を求めた．出力された対応を人手で確認し，再現率，精度，F 値を求めた．ニュースの特徴語ベクトルの大きさは 15 語とした．ニュース記事が書かれて 7 日以内に書かれた blog に対し対応付けを試みた．

表 1: ベクトル空間法の実験手法

	blog 重み	尺度
手法 1	IDF	内積
手法 2	IDF	Cosine
手法 3	TFIDF	内積
手法 4	TFIDF	Cosine

blog は非常に数が多いので，あるニュースに対し言及している blog の真の数を知ることは難しく，再現率は正確な値が分からない．本実験では，実験中に人手で発見できたものを対応付けされる blog の全てと見なして評価を行った．ただしその数が 50 件を超える場合，対応づけられる blog 数は 50 件として評価し，システムが対応付けされると判定する blog の件数の上限を 50 とした．

4.2 使用したデータ

ニュース記事として，朝日新聞，読売新聞，毎日新聞各社の Web サイトから収集したニュース記事を使用した．タイトル，本文と日付を収集している．2004 年 8 月から 11 月に書かれた記事を使用した．

blog として，南野ら [1] が開発したシステム (blogWatcher[1]) により収集された blog を使用した．収集された blog には，本文の他に書かれた日付も含まれている．ニュースと同様に 2004 年 8 月から 11 月に書かれたものを使用した．blogWatcher では blog の文書の単位をトピック単位ではなく，日付単位としている．そのため一つの blog 内に複数の話題が含まれることが多い．

ニュース，blog とも，収集された本文は ChaSen[2] を用いて形態素解析し，名詞，アルファベット，未知語をベクトルに含める語として使用した．

4.3 ベクトル空間法の実験

まず，一般的な TFIDF と Cosine による類似度と比べ，本研究で用いた IDF と内積による類似度が有効であることを示す．

表 1 に比較した手法を示す．提案手法は手法 1 であり，一般的な TFIDF と Cosine による類似度は手法 4 である．表中の blog 重みは，blog の重み付けを IDF のみで行うか，TFIDF を用いるかを意味し，尺度は内積を用いるか Cosine を用いるかを表している．

表 2 に結果を示す．提案手法である手法 1 が最も良い結果を得た．手法 3 は結果が極端に悪くなっているが，これは TF という文書の長さに依存した要素を使っているにも関わらず，長さによる正規化をしていないためである．

[1] http://blogwatcher.lr.pi.titech.ac.jp/
[2] http://chasen.org/

表 2: ベクトル空間法の実験結果

	再現率	精度	F 値
手法 1	0.539	0.796	**0.643**
手法 2	0.474	0.550	0.509
手法 3	0.507	0.152	0.233
手法 4	0.441	0.385	0.411

表 3: 語の出現頻度推移による重みの実験結果

	再現率	精度	F 値
重み付けあり	0.707	0.888	**0.787**
なし	0.539	0.796	0.643

4.4 語の出現頻度推移による重みの実験

次に語の出現頻度推移による重みを用いた実験を行った．

結果を表 3 に示した．語の出現頻度推移による重みを用いることで良い結果が得られていることが分かり，この重みが有効であったと言える．

語の出現頻度推移による重みは，語の出現頻度が大きく変化した時のみ重みを付けるため，言及する blog の多いニュースほど良い結果を得ると考えられる．しかし今回の実験では全ての対応付けを人手で見る必要があったため，システムが対応付けする blog を 50 件に絞っていた．語の出現頻度推移による重みは，それが 50 件を超えるようなニュースにこそ有効であると考えられる．

そこで，対応付けすべき blog が 50 件を超えるニュース 1 件に対し，システムが対応付けする blog の上限を 200 件とし，表 3 の値を得た際の閾値を用いて F 値を求めた．

その結果を表 4 に示した．語の出現頻度推移による重み付けの有無で F 値に大きな差がでた．重み付け無しの場合，対応付けする blog を 50 件に絞っていたためにこれまでのような数値になっていたが，真の値はずっと低い値であったと考えられる．重み付けをした場合でも F 値は低下しているが，重み付けなしの場合ほどは低下しておらず，この重み付けが有効に働いたと考えられる．

4.5 Term Distillation を用いた実験

最後に 2 節で述べた関連研究である Term Distillation を用いた実験を行った．クエリデータベースとしてニュース記事を，検索対象データベースとして blog を用いる．ニュースの特徴語ベクトルを生成する際に，各語の TDV を求め，TDV の高いものから 15 語を特徴語ベクトルとして使用する．Term Distillation を使用しない場合は 4.3 節の手法 1 と同じ手法である．

結果を表 5 に示した．Term Distillation をした結果，しない場合に比べて悪い結果となった．

Term Distillation は，ニュース記事をクエリとして特許を発見するというタスクでは成果を挙げていた．特許の文書には，ニュース記事に比べ，専門的な単語が多く登場する．そのため，ニュース記事からクエリとして使用する必要のある語は，ニュースではあまり使われず，特許文書にだけ頻繁に現れる語である．このような場合には，Term Distillation が有効であると考えられる．

表 4: 対応付けする上限を 200 件とした実験

	再現率	精度	F 値
重み付けあり	0.410	1.000	**0.582**
なし	0.010	1.000	0.020

表 5: Term Distillation の実験結果

	再現率	精度	F 値
Term Distillation あり	0.349	0.525	0.419
なし	0.539	0.796	**0.643**

特許の代わりに blog を発見する場合，Term Distillation によって抽出されるのはニュースにはあまり使われず，blog にだけ頻繁に現れる語である．blog には日常的なことが書かれることが多く，逆にニュース記事には日常的なことは書かれない．つまり，Term Distillation によって抽出される語は日常的な語が多く，ニュース記事との対応付けには不要な語が選ばれてしまったと考えられる．

5 おわりに

本稿では blog とニュース記事間の対応付けが，それらの性質の違いを考慮しつつ行うことで，性能を向上させることができることを示した．

実験によって，ニュース記事と特許の間では使える技術がニュース記事と blog の間では必ずしも使えないことが分かった．またニュースに影響されて blog に出現する語の頻度が変化する，という性質が，ニュース記事との対応付けをする際に有効であるということも分かった．

今後の課題として，対応付けにニュース記事のクラスタリングを利用することを考えている．同じ内容のニュース記事には同じ blog が対応づけられるはずである．こういった性質を利用することで，blog とニュース記事の対応付けにより良い結果をもたらすことができると考えている．

参考文献

[1] 南野 朋之, 鈴木 泰裕, 藤木 稔明, 奥村 学. blog の自動収集と監視. 人工知能学会論文誌, 第 19 巻 6 号, pp. 511–520, 2004.

[2] Hideo ITOH, Hiroko MANO, Yasushi OGAWA. Term Distillation for Cross-DB Retrieval, *Working Notes of the 3rd NTCIR Workshop Meeting, Part III : Patent Retrieval Task*, pp. 11–14, 2002.