ニュース記事のための対話的トピック分析システムとその高速化手法について

平田 紀史† 浅見 昌平† 大囿 忠親† 新谷 虎松†
名古屋工業大学大学院工学研究科情報工学専攻†

1 はじめに

ニュース記事を読む場合に，記事に関係する過去の情報を知りたいということがある．これに対し，例えば Web 上のニュース記事の場合，記事の末尾に関連記事として過去のニュース記事を複数提示することがある．しかし，この場合は関連記事を閲覧することを促しており，閲覧すれば正確な理解を得ることはできるが，時間を消費する．また，ニュース記事は大量に配信されており，実際に関連しているすべてのニュース記事を読むことは困難である．そこで，そのトピックに関連する記事をサブトピックとして内容ごとに分類し，サブトピック間の関係を示すことができれば，トピックの把握が容易になると考える [1].

本論文では，ニュース記事の理解を促すためのトピック分析システムとその高速化手法について述べる．また，本研究ではトピックを特定のキーワードを含む記事集合とする．

2 対話的トピック分析システム

対話的トピック分析とは，提示された結果からキーワードを修正し，繰り返しシステムに問い合わせて，トピックを分析することである．これは，ユーザが望むトピックを一度の要求で提示することは困難であるからである．ユーザは興味のあるキーワードを対話的トピック分析システムに入力し，システムはそのキーワードに関するトピックについて分析し提示する．そして，ユーザが想定しない内容が提示された場合は，入力するキーワードを追加，変更し再度分析を行う．システムの内部では，あらかじめ収集しておいた記事からキーワードに合致する記事を検索し，出現頻度やクラスタリングを行う．収集した記事は前処理として形態素解析や tf-idf 値の計算を行っておく．このような対話的な分析を実現するシステムにおいては短い応答時間が望まれる．

実際のトピック分析の結果は図 1 のようにユーザに提示される．図 1(1) にトピックに属する記事の配信頻度，(2) にトピック内での単語の評価値の変化，(3) にサブトピック間の関係を表す図が提示される．図 1(1)，(2) は数え上げるだけで提示できるため，処理時間は短い．本論文では図 1(3) における分析の高速化を図ることになる．

トピックを分析するに当たり，記事をバッチ的に分析する場合は，分析時間は大きな問題ではない．しかし，ニュース記事は高頻度で配信されており，最新の記事も含めた分析を行おうとすると，逐次的に分析を行う必要がある．サブトピックを得るために，単純に階層的クラスタリングを行うと，記事数によってシステムの応答時間が長くなる．そこで，本論文では，同じ程度の分析結果を出力しつつ，分析時間が短くなることを目標にした手法を提案する．

3 対話的トピック分析のためのクラスタリングについて

階層的クラスタリングは，クラスタ数の変更が容易という利点がある．しかし，計算量は記事数を N とすると $O(N^2)$

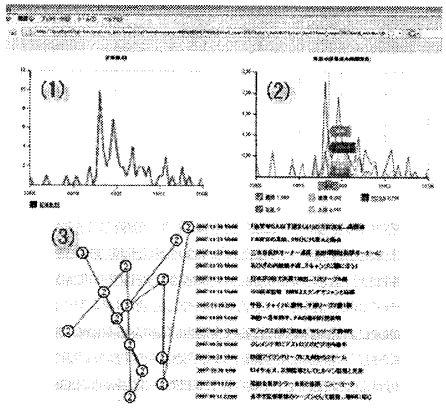

図 1: 分析結果の提示

であるため，トピック分析で用いると分析時間が長くなる．そこで，分析時間を減少させるための手法を提案する．

階層的クラスタリングにおいては類似クラスタを探すための比較に時間が掛かる．特にクラスタリングの初期段階で多くのクラスタがある場合，比較回数が増加する．そこで，この段階でのクラスタリングに階層的クラスタリングよりも計算量の少ない手法を用いれば，処理時間は減少する．つまり，階層的クラスタリングにおける計算量である $O(N^2)$ の N 自体を減らすことで高速化を図る手法である．以下にその具体的な手法を記す．

step1 時系列にソートした記事を記事数 L で区切り，初期のクラスタとする．

step2 クラスタごとに自己類似度を求め，閾値 S 以上であれば step3 の処理は行わない．

step3 クラスタを k-means 法で 2 つに分割を行い，自己類似度が閾値 S 以上となるまで繰り返す．

step4 得られたクラスタに対して階層的クラスタリングを適用し，求まったクラスタをサブトピックとする．

step1 では記事数で区切り初期のクラスタとする．対象とする記事はユーザの入力したキーワードによって分野が絞られているため，時間的に近い記事の内容は類似している可能性が高い．また，時間ではなく記事数で区切ることで，以降の step3 では処理時間に上限を設けることになる．しかし，記事数で区切るだけでは異なる内容の記事が同じクラスタに含まれる場合があるため，step2 の処理を行う．

step2 ではクラスタのまとまりの程度を式 (1) で示す自己類似度という指標を用いて判断する．

$$s_i = \frac{1}{N}\sum_{j=1}^{N} cos(i,j) \quad (1)$$

式 (1) においてクラスタ i に含まれる記事の平均ベクトルと，クラスタ i 内の記事 j のベクトルとのコサイン類似度を $cos(i,j)$ と表す．N はクラスタ内の記事数である．この自己類似度は，分散の距離関数をコサイン類似度に置き換えたものに等しい．クラスタに含まれる記事のベクトルが類似している場合は，自己類似度が大きくなり，記事の内容も類似していると判断する．逆に自己類似度が小さい場合は，異なる

An Interactive Topic Tracking System for Browsing News Articles and a Method for Fast Response Time

Norifumi HIRATA, Shouhei ASAMI, Tadachika OZONO and Toramatsu SHINTANI

† Dept. of Computer Science and Engineering, Graduate School of Engineering Nagoya Institute of Technology, 466-8555, Nagoya, Japan

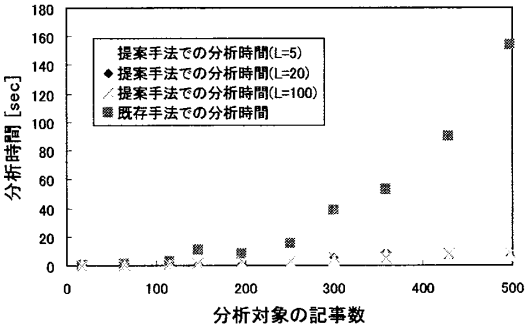

図 2: 記事数とトピックの分析時間の関係

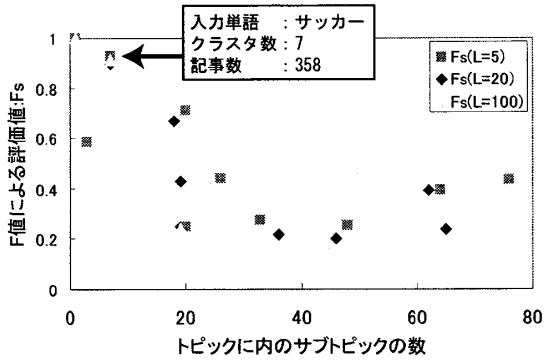

図 3: トピック内のサブトピックの数と F_s の関係

内容の記事が同一クラスタに含まれていると判断し，クラスタを複数に分割する．

step3 では自己類似度が大きくなるように記事を分割する．分割は最小限に抑えるため，k-means 法を用いて 2 つに分割し，分割後のクラスタの自己類似度が閾値 S 以上になるまで繰り返す．クラスタ内において 2 分木を構成することになるが，これが完全に平衡ならば計算量は $O(N\log N)$ 程度であり，高速である．step4 において階層的クラスタリングを行うため，step3 においても同じ手法を用いる方が自然である．しかし，提案手法の目標は高速化であるため，より早い手法を用いる．

最後に，step4 において階層的クラスタリングを行うが，ある程度まとまりのあるクラスタを対象にクラスタリングを行うため，組み合わせ数が減少し処理時間の短縮につながる．また，本論文において，階層的クラスタリングでの距離関数はすべて Ward 法を用いている．

提案手法では step1 において記事を区切る L が重要となる．L が小さい場合は単純な階層的クラスタリングと同等であり，L 大きい場合は k-means 法を用いた分割型の階層的クラスタリングと同等になるためである．

4 評価実験

毎日 jp[1]が 2007 年 10 月 1 日から 2007 年 11 月 30 日まで配信した 10692 記事を収集し，分析を行った．比較するために，トピック分析でのクラスタリングにおいて，単純に階層的クラスタリングを行った場合での分析も行った．以降は単純な階層的クラスタリングを既存手法と呼ぶことにする．実験環境は Windows XP Home Edition，CPU が Pentium M 1.20GHz，メモリが 1GB DDR2 SDRAM，Java の実行環境が JRE1.5.0 である．

4.1 処理時間

3 節で示した L を 5，20，100 と変化させた場合の分析時間と，既存手法を適用した場合の分析時間を図 2 に示す．分析対象の記事数が増加すると単純な階層的クラスタリングの場合では分析時間が多くなるが，提案手法の場合は記事数による変化は少ない．また，L を変化させた場合に，分析時間は L が大きい場合の方が若干多くなった．L を大きくすると 3 節の step3 における分割回数が増加するが，step4 でのクラスタリングの比較回数は少なくなる．このトレードオフの関係が L を変化させても分析時間の変化が少なかったことの原因と考えられる．

4.2 クラスタリング結果の比較

提案手法を用いて，階層的クラスタリングを適応した場合との比較を行う．その手法として F 値を用いた評価式を用いる．F 値は適合率と再現率の調和平均である．クラスタリング全体の評価は，既存手法でのサブトピックごとに最大となる F 尺度を求め，文書数で重み付けした値 F_s[2] とする．

$$F_s = \sum_h \frac{\tilde{n}_h}{N} \max_k F_m(K_h, C_k) \qquad (2)$$

F_m は単純な階層的クラスタリングから得られたクラスタが K_h，提案手法で得られたクラスタが C_k の F 値である．また，N はトピックに属する記事数で，$\tilde{n}_h$ は K_h に含まれる文書数である．

L を 5，20，100 と変化させた場合の F_s を図 3 に示す．記事数と F_s に相関関係が少なかったため，横軸はクラスタ数とした．図 3 から F_s はトピックに属するクラスタ数に関係があることが確認できた．F_s が 1 である結果については，これはサブトピックを示すクラスタが 1 つになっていることが原因である．図 3 では，クラスタ数が 20 前後で急激に F_s が小さくなる．これにより，クラスタ数 20 前後に既存手法とは異なる結果を出力する境界があると考えられる．“サッカー” のように記事数が多い場合でもクラスタ数が少ない場合は F_s 値が高くなることが確認できた．したがって，提案手法では分析対象の記事数ではなく，クラスタ数が少ない場合に，既存手法を用いた手法と類似した結果が得られることになる．しかし，既存手法の結果が本来望むサブトピックであるとは限らないため，クラスタ数が多い場合に結果が悪くなると一概に判断することはできない．

5 おわりに

実験の結果，提案手法ではサブトピックを表すクラスタ数が少なければ，既存手法を用いた場合と同程度の結果を得ることができた．“経済” や “大学” など，一般的なキーワードの場合はクラスタ数が多くなることがある．クラスタ数が多くなると，本来の目的であるトピックの変化の把握には適さない．そこで，ユーザは何度かシステムに問い合わせてトピックを絞っていく必要がある．しかし，提案手法により，分析時間は小さくすることに成功したため，その作業は容易になった．

本論文では，k-means 法，Ward 法を用いたが，今後は他手法での結果の違いについても検討する必要がある．

参考文献

[1] 平田紀史，大囿忠親，新谷虎松: エージェントによる滑走窓方式を用いた複数情報源からのトピック追跡，合同エージェントワークショップ & シンポジウム 2007, CD-ROM, 2007.

[2] Bjornar Larsen and Chinatsu Aone : Fast and effective text mining using linear-time document clustering, Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining, 1999, pp.16-22.

[1] http://mainichi.jp/