記事集合における確率的手法を用いた話題の発見と追跡

北村佑介 [1*]　村田剛志 [2]

[1] 東京工業大学工学部情報工学科
[2] 東京工業大学大学院情報理工学研究科計算工学専攻

Abstract: 本研究では Topic Detection and Tracking(TDT) の技術を用いて、既存の話題の追跡や話題の発見を行い、記事集合から話題構造を抽出した。話題構造を抽出できれば、Web 上の膨大な数の記事に対して、ユーザの効率的なアクセスをサポートすること等に応用できる。記事と話題との類似度を確率的な手法で計算することで、記事を話題に分類したり新規話題かを判断したりする。さらに複数の類似度の計算手法を使用し、それらを比較する。実験の結果、コーパスと異なる粒度で話題が形成されたために精度は高くなかった。また各計算手法は、話題追跡に適するものと話題発見に適するものとで 2 種類に分かれた。以上の結果を、記事データの特徴や話題の粒度との関係をふまえて考察した。

1　はじめに

本研究では、Topic Detection and Tracking(TDT)[1] と呼ばれる技術を用いて、日本語の記事集合に対して話題の発見と追跡を行うシステムを構築しその有効性を調べた。話題の発見とは、新しい話題やそれまで認識していなかった話題を検出することであり、話題の追跡とは、既知の話題に対してその話題と関係のある記事を分類することである。Web 上でユーザの興味のある話題を追跡することで、その話題に分類された記事をユーザに提供することが可能になる。さらに新しい話題を発見することで、流行情報をユーザに提供し、ユーザに対してより多くの話題の中から興味のある話題を選んでもらうことにも繋がる。システムでは、日本語で記述された未分類の記事集合をテスト記事として入力し、その記事と既知の各話題との類似性を確率的な手法を用いて計算する。実験では類似度の計算手法として 4 通りを試して精度を比較すると共に各手法の特徴を考察する。類似度を正規化した後、テスト記事が最も高い類似性を示す話題を探して、その類似度が閾値を越えていればその話題に分類する。どの話題との類似度も閾値を越えないなら新規話題としてみなし、記事を新しい話題の核とする。

システムを実装し、類似度や閾値を変えて話題の発見と追跡を行った。システムの精度が正解とするデータに依存するために、結果の精度は良好なものではなかったが、こちらが予測していなかった話題の発見と追跡も行われていた。また、4 つの類似度の計算手法が発見に適した手法と追跡に適した手法に分かれることが確認できた。この結果を話題の粒度や実験に使用した情報との関係をふまえて考察を行った。

*連絡先：東京工業大学工学部情報工学科
住所：東京都目黒区大岡山 2-12-1 W8-59
E-mail：kitamura03@ai.cs.titech.ac.jp

2　話題の発見と追跡

ここではシステムで使用する話題の発見と追跡の手法を述べる。

2.1　システムの流れ

話題の発見と追跡を行うシステムの概要を図 1 に示す。

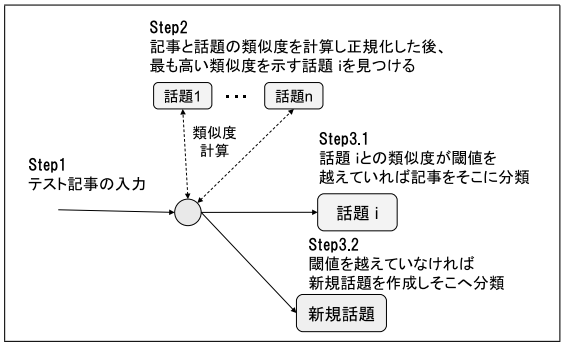

図 1: システム概要

まず、システムに対してテスト記事を入力する。次に、テスト記事と既知の話題との類似度を計算する。類似度の計算手法は 2.2 で述べる。これらを比較して最も高い類似度を示す話題 t を見つける。しかし、その前に類似度は話題ごとに正規化を行う必要がある。なぜなら話題によって、それに含まれる記事数などの条件が異なることで類似度は大きく異なってしまうからである。正規化の方法は、話題 t との類似度を正規化するとした場合、話題 t に含まれない記事群 (話題 t に関して off-topic 記事) と話題 t との類似度を計算して、その平均 μ_{OFF} と偏差

σ_{OFF} を求める。そして以下の式に代入する。

$$score_N(s,t) = \frac{score(s,t) - \mu_{OFF}}{\sigma_{OFF}}$$

正規化された類似度を比較して、最も高い類似度を示す話題 t を見つけておく。その後、テスト記事と話題 t との類似度がある閾値を越えているかを調べる。閾値を越えていれば、テスト記事は話題 t に分類される。逆に閾値を越えていなければ、テスト記事は既知のどの話題とも類似度が低いということになり、新規話題を作成して記事をそこへ分類する。また、閾値はテスト記事と関連のある記事との最低限のラインということになるが、実データによってその閾値は異なるために実験の中で複数回試す必要がある。

テスト記事の分類が終了した後、再推定を行う。再推定では、それまで分類したテスト記事を再検査する。テスト記事が追加されることによって新規話題が作成されたり話題の追跡が起こり、全体の状況が変化する。そこで、すでに分類されたテスト記事と各話題との類似度を再度計算して、より高い類似度を示す話題があった場合にそちらへ記事を移動させる。

2.2 類似度

本研究では、4 つの類似度の計算手法を用いて比較を行う。ただし、確率を計算する際に 0 となるような場合にはディスカウンティング [2] を行う。

Topic Spotting(TS) measure

TS measure[4] は、記事 s がその話題 t と関連がある条件付確率を類似度とみなす。計算式は、

$$M_{TS}(s,t) = \log \frac{P(s|t)}{P(s)} \approx \sum_{w \in s} \log \frac{P(w|t)}{P(w)}$$

となる。ここで、$P(w|t)$ は話題 t における単語 w が出現する事前確率、$P(w)$ はコーパス全体で単語 w の出現する事前確率である。

Information Retrieval(IR) measure

IR measure[4] は、話題 t がその記事 s と関連がある条件付き確率を類似度とみなす。計算式は以下のように表せる。

$$M_{IR}(s,t) = \sum_{w \in t} \log(aP(w|s) + (1-a)P(w))$$

ここでは話題 t における単語 w が関連のある記事による言語モデル ($= P(w|s)$) とコーパス全体による言語モデル ($= P(w)$) の 2 つのモデルによって生成されたとみなすことで精度の向上を狙っている。このような 2 つの言語モデルからなるような言語モデルは隠れマルコフモデル [2, 3] と呼ばれている。また、この 2 つのモデルを混合させる際の重み a は EM アルゴリズムを用いて推定する。

IR with Relevance Feedback(RF) measure

RF measure はほとんど IR measure と同じであるが、ただ一点重み a の決定法が異なる。重み a は単語 w を話題 t と関連のある記事がどの程度使っているかに基づいて計算する。

$$a(w) = P(S\ has\ w|S\ on\ t) - P(S\ has\ w)$$

$$P(S\ has\ w|S\ on\ t) = \frac{N(S\ has\ w, S\ on\ t)}{N(S\ on\ t)}$$

$$P(S\ has\ w) = \frac{N(S\ has\ w)}{N(S)}$$

それぞれ $N(S\ has\ w, S\ on\ t)$ は話題 t に含まれる記事でかつ単語 w を含む記事数、$N(S\ on\ t)$ は話題 t に含まれる記事数、$N(S\ has\ w)$ はコーパス全体で単語 w を含む記事数、$N(S)$ はコーパス全体の記事数を示している。ただし、この場合の $S(\neq s)$ は任意の記事ではなく、一般的な記事のことを指していることに注意する。

Word Presence(WP) measure

WP measure は、話題に対して各単語がどの程度必要条件であるかを数値化し総和をとったものである。必要条件かは、その話題の記事の多くが使用している単語で判定する。式は、

$$M_{WP}(s,t) = \sum_{w \in t} \log \frac{P(S\ has\ w, S\ on\ t)}{P(S\ has\ w)}$$

で表される。

2.3 評価基準

システムによって得られた話題構造の評価は、検出失敗率と誤検出率を用いて行う [5]。まず、誤検出率であるが、これは記事を本来検出するべき話題ではなく誤った話題に検出してしまう確率であり、「誤って検出した記事数」を「本来検出するべきでない記事 (Non-Target Story) の数」で割ったものがこれにあたる。次に、検出失敗率は、記事が本来検出されるべき話題に検出できなかった確率のことであり、「検出できなかった記事数」を「本来検出するべき記事 (Target Story) の数」で割った値とする。これらの値はコーパス (正解) の話題ごとに計算して平均化したものをシステム全体の値とみなす。

3 実験

前節で述べた手法を用いてシステムを構築し、話題の発見と追跡を行った。類似度の計算手法は TS、IR、RF、

WP の 4 つを使用し、それぞれの精度をみるとともに手法間で比較する。また、記事を既知の話題に分類する際に使用する閾値は、適当な値がわからないために 6-7 通りで試した。ただし、基本的には正規化された類似度の平均である 0 を超えるように閾値を設定した。システムの流れや類似度の計算手法は前節で述べた通りなので省略する。ここでは、データセットの作成方法と記事の事前処理について述べる。

3.1 データセットの作成方法

MuST のコーパスを訓練データとテストデータに分けて実験を行う。訓練データ中の話題構造は既知のものとして、テストデータに含まれる記事をシステムに入力して話題の発見と追跡を行う。

先行研究である TDT では、コーパスを時系列で 3 つに区切り、新しい方の 1/3 をテストデータ、古い方の 2/3 を訓練データとしていた。しかし、使用する MuST のコーパスでは、新規話題の発生が時系列の古い期間に集中し新しい方の期間では新規話題が発生しない。従って TDT のような時系列で区切るような方法を用いた場合に新規話題の発見について検証することができない。そのために、次のようにデータセットを作成する。

まず、MuST のコーパスに含まれる 27 のうちの 1/3 にあたる 9 の話題をテストデータに追加する。次に、残りの 18 の話題に関しては、各話題に含まれる記事の 1/3 をランダムに選択してテストデータに追加する。テストデータに追加されなかった記事はすべて訓練データとする。このようなデータセットを 3 種類作成して、その 3 つを同じ条件 (類似度の計算手法と閾値) でシステムに入力して得られた結果を交差検定する。

3.2 記事の事前処理

各記事をシステムに入力する前にいくつか処理を施す。まず、形態素解析を行うプログラム Sen を用いて文章を単語に区切る。ストップワードとして機能語を削除する。機能語は、1 つの単語では意味をなさないような抽象的な単語、もしくは品詞を指す。具体的には、助詞、助動詞、形式名詞、接続詞、連体詞、副詞、記号、数字を機能語とみなす。ストップワードを除去した後、どの単語が何回出現したかをあらかじめ集計しておく。

4 考察

4 つの類似度の計算手法を使用した結果の考察を述べる。まず、TS measure を使用した場合の結果について述べる。この時の特徴は 2 つある。1 つは新規話題の発見が全く行われなかったことである。新規話題とみなされて話題が形成されるのだが、その話題に対してそれ以降の記事が分類されない。つまり、記事数 1 の新規話題しか形成されない。この類似度は、記事の単語がその話題内で使われる確率の総和を取っている。しかし、記事数が 1 つの話題だとその話題内で使われている単語も既存の話題に比べて非常に少ない。そのために、記事数 1 の話題に対して大きな類似度を示さなかった。次に、粒度の大きな話題が形成されたことである。コーパス中の 2 つ以上の話題が統合されることによって 1 つの大きな話題が形成された。例えば、「台風」と「地震」に関する話題が統合することによって災害関係の話題が形成された。検出失敗率と誤検出率を表 1 に示す。結果的に、新規話題の発見ができないことと、こちらの期待する粒度での話題の追跡ができなかったことで検出失敗率が高くなり、精度としてはあまり高くなかった。

閾値	検出失敗率	誤検出率
0.25	0.6661	0.0289
0.50	0.6425	0.0283
0.75	0.6826	0.0270
1.00	0.7155	0.0244
1.25	0.7557	0.0201
2.00	0.8318	0.0097

表 1: TS measure の検出失敗率と誤検出率

次に、IR measure を使用した場合であるが、こちらはある程度の新規話題の発見が可能である一方で、話題の追跡がほとんど行えなかった。新規話題の発見に関しては記事数の少ない話題に対しても記事が分類された。しかし、元は 1 つの話題であったものがいくつかに分裂するということが起こっており完全なものとはいえない。また、話題追跡も既存の 1 つの話題に対してテスト記事が集中して分類されることが起こった。検出失敗率と誤検出率は表 2 に示す。話題追跡ができなかったことが原因で精度が低かった。ただし、誤検出率が低く、新規話題に関して誤って異なる話題の記事が分類されることがなかったことに起因する。

閾値	検出失敗率	誤検出率
0.2	0.9026	0.0104
0.4	0.8991	0.0063
0.6	0.8852	3.165E-4
0.8	0.9118	2.414E-4
1.0	0.9205	2.053E-4
1.2	0.9312	1.574E-4

表 2: IR measure の検出失敗率と誤検出率

RF measure は基本となる式が同じなので IR measure

と似たような結果が得られた。話題追跡に関してはやはり期待通りには行われず、1 つの話題に対してテスト記事が集中することがみられた。また、新規話題発見に関しては行えたものの、IR measure に比べると、話題の細分化が進んでいた。検出失敗率と誤検出率は表 3 に示す。話題追跡ができなかったことと、新規話題の発見に関して上手く行えなかったことが原因で精度が低かった。

閾値	検出失敗率	誤検出率
0.2	0.8971	0.0110
0.4	0.9136	0.0098
0.6	0.9233	0.0078
0.8	0.9271	0.0065
1.0	0.9277	0.0047
1.2	0.9314	0.0027

表 3: RF measure の検出失敗率と誤検出率

WP measure を使用した場合であるが、これは TS measure の結果と類似していた。新規話題発見ができないが、話題の追跡を行うことができた。新規話題の発見ができないのは TS measure と同じ原因である。一方の話題追跡に関してであるが、TS measure に比べるとその精度は低かった。これは、WP measure がバイナリモデルを利用していることが 1 つの原因だと考えられる。WP measure では、その話題に含まれる記事でその単語が複数回使われていても、1 回にカウントされる。このことによって、記事中での出現回数が 10 回の単語と 1 回の単語で差が現れない。検出失敗率と誤検出率を表 4 に示す。表を見てもわかる通り、話題の追跡の精度も高くなかったために精度は TS measure に比べて全体的に低かった。

閾値	検出失敗率	誤検出率
0.25	0.7331	0.0297
0.50	0.7485	0.0268
0.75	0.7793	0.0223
1.00	0.8406	0.0114
1.25	0.8767	0.0090
2.00	0.9271	0.0047

表 4: WP measure の検出失敗率と誤検出率

最後に、各手法を用いた場合の話題の発見と追跡の適性を表 5 にまとめる。TS measure と WP measure は話題追跡に適しており、IR measure は話題発見に適している。RF measure も話題追跡より話題発見に適しているが IR measure に比べると精度が劣った。

	話題発見	話題追跡
TS	×	◎
IR	○	△
RF	△	△
WP	×	○

表 5: 手法の適性

5 おわりに

今回、日本語の記事集合に対して、確率的に類似度を計算することで話題の追跡と分類を行った。全体的に、精度は高くなかった。しかし、手法によって話題発見に適したものと話題追跡に適したものがあることがわかった。また、こちらの期待した粒度で話題が追跡される一方で、複数の話題が統合することで粒度の大きな話題を形成することが確認できた。

今後の課題としてはやはり精度の向上が挙げられる。まず、類似度の計算手法の改善や新たな計算手法の提案がある。また、必要な用途によって使い分けをすることも考えられる。さらに、大きな粒度が形成されることを利用して、階層的に話題の発見と追跡を行い、期待する粒度で話題の検出を行うことも可能だろう。その他、ストップワードをどう選択するか等、類似度の計算に影響を与えるような要素を考慮する必要がある。

また、今回補いきれなかった話題構造の可視化も今後の課題といえる。得られた話題構造をユーザに示す手法がまだ確立されておらず、課題の 1 つといえる。

参考文献

[1] James Allan:TOPIC DETECTION AND TRACKING Event-based Information Organization, KLUWER ACADEMIC PUBLISHERS,(2002)

[2] 北 研二:確率的言語モデル, 東京大学出版会,(1999)

[3] David R. H. Miller, Tim Leek, Richard M. Schwartz.:A hidden Markov model information retrieval system,ACM Press,214-221(1999)

[4] Hubert Jin, Rich Schwartz, Sreenivasa Sista, Frederick Walls:Topic Tracking for Radio, TV Broadcast, and Newswire

[5] Jon Fiscus, George Doddington, John Garofolo, Alvin Martin:NIST's 1998 Topic Detection and Tracking Evaluation (TDT2),(1999)