続報記事抽出のための記事間類似度を利用した SVM 学習データの自動生成

北條 博之， 鈴木 良弥
山梨大学工学部コンピュータ・メディア工学科

1. はじめに

インターネット等の普及により，膨大で多様な情報や文書がオンラインで流通している．それらの大量の情報から必要な情報のみを人手によって取得することは非常に困難である．そのため，「新潟中越地震」や「スマトラ沖地震」等の，ある出来事に関する一連の記事から効率よく知識を得るための技術として続報記事抽出が脚光を浴びている．続報記事抽出の利点として，大量の記事の中から同一の出来事に関する記事のみを取得できるため，その出来事がどのように展開されていったかが分かる等が挙げられる．

続報記事抽出のため，様々な手法が提案されている[1][2][3]が，その中でも，Support Vector Machines(SVM)等の機械学習を利用した手法[4][5]により，高精度な抽出結果が報告されている．続報記事抽出の難しさの一つに学習データ中の正例（当該出来事の記事）の数が負例（当該出来事以外の記事）に比べて非常に少ないことが挙げられる．従って，どの負例記事を学習データとして選択するかが問題となる．また，本来は，ある出来事の続報記事抽出を行う場合，抽出する範囲より過去の当該出来事の記事を使用して，学習データが作成される．そのため，過去の当該出来事の記事が分かる出来事のみでしか，続報記事抽出できない．本稿では，出来事の違いによらず，SVM による続報記事の抽出精度を向上させるためにはどのような学習データが必要か調べ，その結果を基に，正例と負例を含めた学習データの自動生成手法を提案する．（これ以降，「出来事」を「トピック」と表記する．）

2. 本続報記事抽出システムの流れ

事前に，記事を特徴ベクトル（記事の特徴をベクトルで表したもの）に変換することで，SVM での学習や分類，そして，類似度計算ができる．よって，特徴ベクトルを用いて，続報記事抽出を行う．

図 1 に本続報記事抽出システムの流れを示す．本システムは，元記事（取得したいトピックのある記事）を入力するだけで学習データ生成から続報記事抽出までを自動的に実行する．よって，事前には，続報記事を抽出したい範囲の記事を特徴ベクトルへ変換しておくだけで，他の作業（学習データをあらかじめ用意する等）は，行なわない．また，特徴ベクトルに変換した記事の範囲であれば，元記事と同一トピック記事をすべて抽出できる．

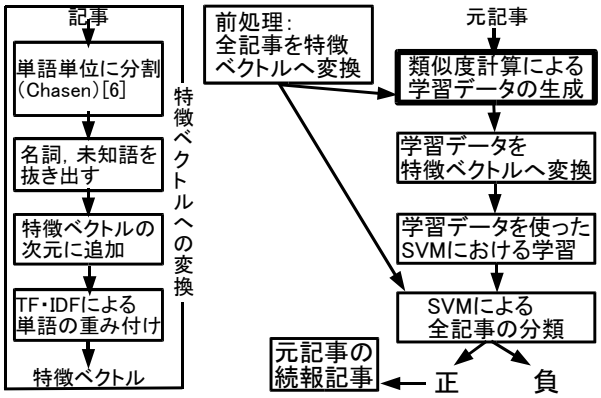

図1：本続報記事抽出システムの流れ

2.1 TF・IDF

記事を特徴ベクトルへ変換する際の単語の重み付けとして，以下の TF・IDF 値を用いる．

$$tf(t)=\frac{\text{文書内に出現する単語}t\text{の数}}{\text{文書内に出現する総単語数}}$$

$$idf(t)=\log\left(\frac{\text{文書総数}}{\text{単語}t\text{が出現する文書総数}}\right)$$

$$TF\ \cdot\ IDF(t)=tf(t)\times idf(t)$$

2.2 類似度計算手法

内積，余弦，Dice 係数，Jaccard 係数[7]といった 4 種類の類似度計算手法を試した．その中で，システムの性能が最も良かったのが，Dice 係数類似度であった．よって，学習データを生成する際や比較実験の際に以下の Dice 係数類似度を用いる．（x_i, y_i はそれぞれ，文書 d_x, d_y の索引語 i に対する重み，T は索引語の総数である．）

$$Sim(d_x,d_y)=\frac{(2\sum_{i=1}^{T}x_i\cdot y_i)}{((\sum_{i=1}^{T}x_i^2)+(\sum_{i=1}^{T}y_i^2))}$$

$$0\le Sim\le 1$$

2.3 SVM(Support Vector Machines)

SVM（Supprot Vector Machines）は，学習データ中で最も他クラスと近い位置にあるもの（サポートベクトル）を基準として，その距離（マージン）が最も大きくなるような位置に識別境界を設定する．(図 2)[8]

本稿では，SVM ツールの SVM^{light}[9](コーネル大学)を使用する．

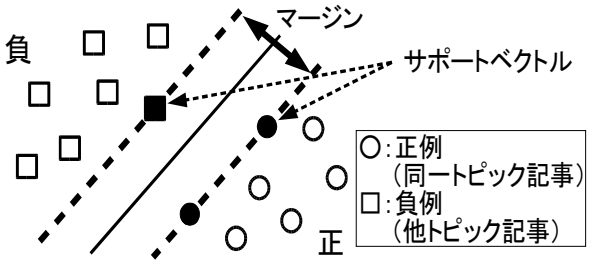

図 2:SVM の概念図

3. 学習データの自動生成

実験データは，ある程度，線形分離可能（正か負かをある位置を境に分離可能）として考え，SVM による識別の前処理として，確実に正例と負例を判断できる学習データを選び出す．

同一トピック記事は少ないため，学習データ中の正例となる記事は，選ぶことはできない．よって，学習データ中の正例には，確実な同一トピック記事を選びたい．そこで，確実な同一トピック記事を選べる条件に設定するための実験を行う．

同一トピック記事以外の記事は，同一トピック記事に比べて極端に多い．よって，学習データに使われる負例を選ぶことで，性能を向上させることができると考えた．負例のサポートベクトル（SV）が正例に近づき過ぎる（正例と似すぎる）と同一トピック記事以外の記事が入ることは少ないが，同一トピック記事を見逃してしまうため，性能が悪くなる．また，負例のサポートベクトル（SV）が正例から遠ざかり過ぎる（正例と全く似ていない）と同一トピック記事を見逃すことは少ないが，同一トピック記事以外の記事が多く入ってしまうため，性能が悪くなる．よって，バランスを考え，適度に正例と似ている負例を見つけることが重要である（図 3）．そこで，最良の負例を見つける実験を行う．

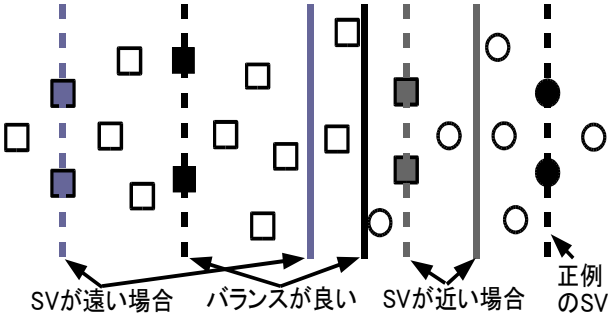

図 3:サポートベクトルの位置による識別境界（点線:SV の位置，実線:識別境界，○:正例，□:負例）

3.1 実験データと評価尺度

続報記事抽出の実験データは，毎日新聞の 1998 年 10 月から 12 月までの 30,777 記事を使用する．

学習データとなる記事も実験データと同範囲とする．よって，学習データ中で正例とした記事が，最終的に続報記事抽出された記事に含まれる．

評価データは，TDT3[10]コーパスの 1998 年 10 月から 12 月までに含まれる 60 トピックについて，毎日新聞の同時期に報告されている 21 トピックを抜き出し，この 21 トピックについて，TDT の評価基準を基に人手で記事の選定を行っている（全 487 記事）．また，今回，実験での元記事には，各トピックの第 1 報記事（各トピック内で日付が最も早い記事）を用いる．

評価尺度は，以下の Precision（精度），Recall（再現率），F 値（総合値）を用いる．

$$Precision = \frac{\text{システムが正確に分類した記事数}}{\text{システムが正例と判断した記事数}}$$

$$Recall = \frac{\text{システムが正確に分類した記事数}}{\text{人が正例と判定した記事数}}$$

$$F\text{値} = \frac{2 \times Recall \times Precision}{(Recall + Precision)}$$

3.2 学習データ中の正例の選び方

類似度の閾値を変化させ，確実な同一トピック記事が多く含まれる度合い（F 値）を比較する．閾値が 0.3 の時，最も良くなった．（図 4）

同一トピック記事数が少ないのに，正例数が多くなってしまうと，正例に同一トピック以外の記事が多く入ってしまうため，正例数の上限を決める．元記事との類似度上位記事数を変化させ，確実な同一トピック記事が多く含まれる度合い（F 値）を比較する．元記事との類似度上位記事数が 15 記事の時，最も良くなった．（図 5）

実験結果より，学習データ中の正例の条件は，元記事との類似度 0.3 以上かつ上位 15 記事以内と設定した．（正例数が少なすぎると，分類性能が悪くなるため，学習データ中の正例数が 5 記事未満になる場合は類似度上位 5 記事を正例とする．）

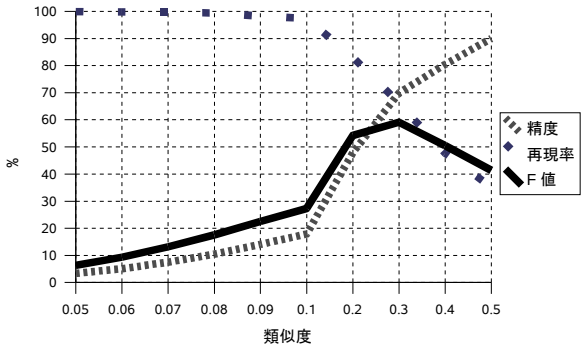

図 4:元記事との類似度による同一トピック記事の抽出

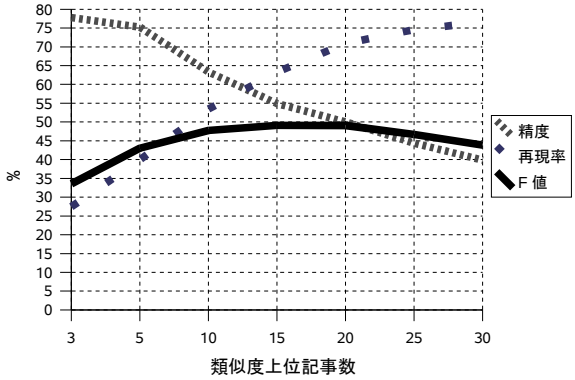

図 5:元記事との類似度上位による同一トピック記事の抽出

3.3 学習データ中の負例の選び方

元記事との類似度がどの程度の負例を用いると性能が最も良くなるかを調べる. 具体的には, 負例数を正例数の 2 倍に固定し, 元記事との類似度を変化させて, 性能を比較する.(正例は, 3.2 節で設定した条件で生成する.)類似度が 0.1 付近の負例を用いた時に, F 値が最も良くなった.(図 6)

次に, どの程度の負例数があれば, 性能が最も良くなるかを調べる. 負例数は多すぎると処理時間が長くなり, 効率が悪くなるため, できるだけ少ない負例数にする. 具体的には, 元記事との類似度が小さい記事を使い, 負例数を変化させて, 性能を比較する.(正例は, 3.2 節で設定した条件で生成する.)負例数が正例数の 5 倍の時に F 値がほぼ最高となり, その後ほとんど変化しない状態となった. (図 7, 図 8)

実験結果より, 学習データ中の負例の条件は, 元記事との類似度 0.1 以下かつ正例数の 5 倍と設定した. 学習データ中の正例は, 3.2 節で設定した条件で生成する. これを本手法とする.

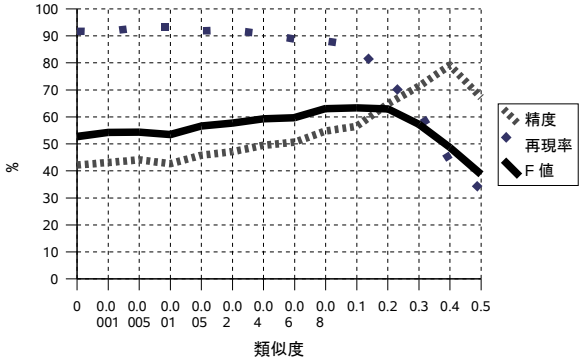

図6:類似度による続報記事抽出性能の比較

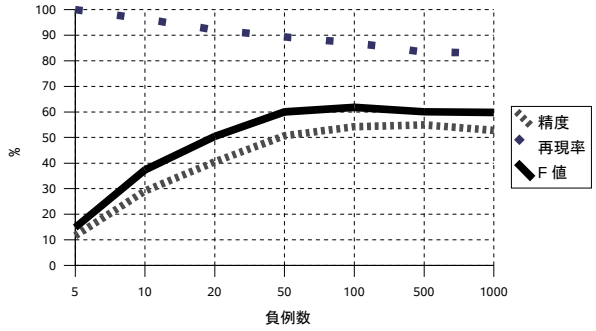

図 7:負例数による続報記事抽出性能の比較 1

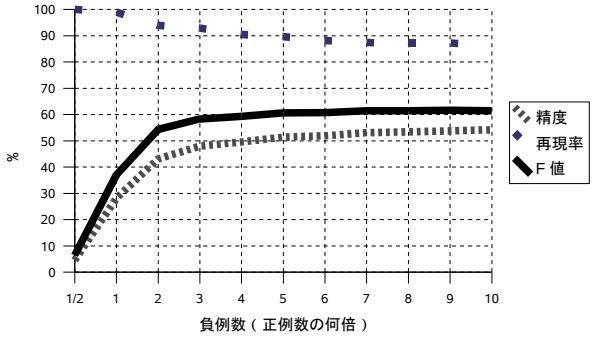

図 8:負例数による続報記事抽出性能の比較 2

4. 続報記事抽出システムの実験

本手法との比較用に学習データ中の正例は同条件で負例をランダムに正例数の 5 倍の記事のもの(ベースライン)と SVM を使わずに類似度計算のみ(閾値 0.3)で続報記事抽出したものを用いる.

4.1 実験結果

表 1 に続報記事抽出実験の結果を示す. 表中の値は F 値, トピック NO.の括弧内の数字は, そのトピック記事数である. 実験の結果, 本手法は, 学習データ中の負例の選び方をランダムにしたベースラインや, 類似度計算のみの続報記事抽出よりも, 性能が良くなることが分かった.

表 1:続報記事抽出システムの結果

トピック ***NO***	本手法	ベースライン	類似度計算
1(26)	0.754098	0.661603	0.653061
2(103)	0.906667	0.884133	0.422535
3(20)	0.952381	0.930233	0.952381
4(4)	0.888889	0.888889	1.000000
5(6)	0.222222	0.110202	0.285714
6(4)	0.545455	0.274423	0.600000
7(7)	0.666667	0.737729	0.666667
8(18)	0.653846	0.586734	0.857143
9(11)	0.818182	0.705806	0.588235

トピック *NO*	本手法	ベースライン	類似度計算
10(63)	0.733945	0.754413	0.375000
11(12)	0.846154	0.888889	0.761905
12(17)	0.619048	0.536527	0.687500
13(1)	0.022727	0.016401	0.125000
14(5)	0.909091	0.909091	0.909091
15(12)	0.631579	0.401400	0.857143
16(16)	0.484848	0.367233	0.500000
17(54)	0.469136	0.624627	0.394737
18(65)	0.554839	0.460169	0.186667
19(23)	0.276730	0.289728	0.291667
20(16)	0.780488	0.765621	0.551724
21(4)	0.888889	0.888889	0.750000
平均値	0.648851	0.603940	0.591246

4.2 考察

本システムの実験結果より，性能が悪くなる原因として，以下のものが挙げられる．

- 学習データ中の正例に同一トピック記事以外の記事が多く含まれる場合，または，学習データ中の負例が正例から遠すぎる場合，識別境界が負例側に寄り過ぎて，性能が悪くなる．
- 学習データ中の負例に同一トピックの記事が多く含まれる場合，または，学習データ中の負例が正例と近すぎる場合，識別境界が正例側に寄り過ぎて，性能が悪くなる．
- 特徴ベクトル空間中で，同一トピック記事同士が正例側に固まっていない場合，SVM での分類性能が悪くなる．

本手法では，学習データ生成条件を一定の値に決めている．そのため，その条件では，続報記事抽出が上手くいかず，極端に性能が悪くなるものが見られた．改善のためには，元記事等から特徴を見つけ，個別の条件設定が必要である．

また，本システムでの，単語の重み付けは，TF・IDF 値を用いているが，より記事の特徴を出し，分類性能を向上させるために，重み付けを工夫していく必要がある．その手法として，以下に挙げたものが有効だと考えられる．

- 新聞記事の特徴として，タイトルや第 1 文に重要な内容が含まれているとされているため，記事の前半の文の重みを大きくする重み付け
- IDF の工夫として，単語の文字数が多い，画数が多い，連接単語等の珍しい単語の重みを大きくする重み付け

5. まとめ

本稿では，記事間類似度を利用して，SVM 学習データを自動生成し，続報記事抽出する手法について述べた．

実験の結果，学習データ中の負例の選び方により，性能を向上させることが可能であることを示した．また，第一報記事以外（全評価データ）が元記事であっても，ほぼ同等の性能（平均 F 値 0.61）となることを確認した．改善の余地はあるが学習データにより多くの正例を用いた続報記事抽出性能[1]（平均 F 値 0.68）とそれほど変わらない性能を得た．

今後は，元記事の選び方やトピックの違いに対して，性能が変化しにくい頑健なシステムにするために，学習データ生成時の条件を個別に決定できるようにすることや重み付けの工夫等が求められる．

参考文献

[1] 奥，鷲崎，田中：「関連記事の判定に関する検討」言語処理学会第 2 回年次大会論文，pp89-92(1996).
[2] 大竹，山本，増山：「名詞の連接に着目した日本語新聞記事の関連記事検索手法」言語処理学会第 3 回年次大会論文，pp381-384(1997).
[3] 山田，金，柴田，浦谷：「ニュース記事を利用したトピック抽出の検討」言語処理学会第 5 回年次大会論文，pp116-119(1999).
[4] Yang, Y., Ault, T., Pierce, T. and Lattimer, C.W.: Improving Text Categorization Method for Event Tracking, *Proc. 23rd Anuual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.65-72 (2000).
[5] Taira, H. and Haruno, M.,: Feature Selection in SVM Categorization, *Proc. Of the Fifth International Conference on Aftificial Intelligence (AAAI-99)*, pp.480-486(1999).
[6] 形態素解析システム Chasen, <http://chasen.aist-nara.ac.jp/hiki/ChaSen/>
[7] 徳永健伸：「情報検索と言語処理」東京大学出版会，pp.31(1999).
[8] 前田英作：「サポートベクトルマシン」情報処理 42 巻 7 号，pp.676-683(2001.7).
[9] SVM ツール SVM[light], <http://svmlight.joachims.org>
[10] TDT3(Topic Detection and Traking Phase 3), <http://www.ldc.upenn.edu/Project/TDT3/>

1 学習データは第一報記事から 1～16 番目の正例，負例を含むすべての記事からなり，実験データはそれ以降のすべての記事とする．その時の続報記事抽出性能である．