SVM と LSTM を用いた政党支持に関するツイート推定

馬 青[†] 花岡 見帆[†] 村田 真樹[††]

[†]龍谷大学大学院理工学研究科数理情報学専攻 [††]鳥取大学大学院工学研究科情報エレクトロニクス専攻

1 はじめに

各マスメディアが発表している世論調査[1]の政党支持率は本当に正しいのだろうか. 世論調査の方法は面接調査, 郵送調査, 電話調査, インターネット調査などが挙げられるが, マスメデイアが最も用いている方法は電話調査の RDD 法である. しかし特定の政党を支持する傾向があるマスメディアが存在することなどから, 多くの問題やバイアスが考えられるため, 世論調査に対して疑念の声が多く上がっているのが現状である.

近年, SNS を用いた政党の支持率や選挙結果の予測を行う研究が多くなされている. 従来の研究では候補者のフォロワー数や, 候補者のツイートへの有権者の「いいね」やリツイートの数に着目した研究が多いが, 最近, Twitter のツイート内容から政党の支持率を予測するという研究がなされている[2]. この研究は Twitter の各政党を含むツイートを対象に感情情報の抽出を行い, 目的変数に政党支持率を定めて重回帰分析を行い, 説明変数の係数を求めることで政党支持率を予測することを可能にした. しかしこの研究ではマスメディアが出した支持率を学習データとして利用していた. また, 検索ワードを含んだツイートを分析しネガポジの比率を推定する Yahoo!つぶやき分析ツールも開発されている[3]. しかしこのツールは辞書方式から機械学習方式に変更したとあるが, 中身が公開されておらずブラックボックスである.

本研究はマスメディアが出した支持率への疑問提起から始めたもので, 個々のユーザーに着目し, 個々のユーザーが発したツイートの政党支持・不支持を判定して支持率を予測することを目的としている. 具体的には, ユーザーの各ツイートの支持・不支持の推定を行い(課題1), その結果からユーザーの政党支持・不支持を判定して支持率を予測する(課題2)ことを目指す.

2 課題設定

本稿は課題1に限定したものである. 具体的には, ツイート内容に政党を含んでいるとしても政党に対して言及していないツイートが多く見られることから,「ネガティブ(批判的)」,「ポジティブ(肯定的)」,「言及していない」の3クラス推定を機械学習で行う. 機械学習に SVM と LSTM(Long Short Term Memory)を用いる. 学習データが必要なため, コーパスを作成する(3節). 機械学習のためのベクトル化については本課題に特化した政党支持・不支持の観点から, 反対勢力を批判した造語(例:ネトウヨ, パヨク)なども考慮した複数の方法を考案する(4節). 提案手法の有効性を確認するために, 複数の方法で構成したベクトルを用いた SVM と LSTM 間の比較実験(6節), さらにはベースライン(5節)として作成した手法との比較実験を行う.

3 コーパスの作成

Twitter API[4]を用いて 2018 年 4 月 30 日から 10 月末までの政党を含むツイートデータ(ツイート ID・アカウント名・ツイート日時)を取得した. ツイート取得用のキーワードは与党から自民党と社民党, 野党から主要2党の民主党と共産党の4種である. 教師あり学習データ(タグ付きコーパス)を作成するためにツイートデータにタグを付与する必要があった. タグの種類を「自民党」,「社民党」,「民主党」,「共産党」,「与党」,「野党」,「右翼」,「左翼」とし, さらにこれらに対して「ネガティブ(批判的)」,「ポジティブ(肯定的)」,「言及していない」という3種類のタグを設けた. 政党以外のタグ(つまり「与党」「野党」「右翼」「左翼」という4種類のタグ)はどのようなユーザーがどのような政党を支持しているかに関係していると考え, 今後支持率を予測する課題2において必要に応じて利用するため用意した. 本稿では「自民党」に限定し,「自民党」を含むツイート 2000 件にタグ付けを行った. また, ツイートデータにおいてひらがな・カタカナは全角, そして英数字は半角の大文字に統一した. さらにツイートの推定に不要な情報である URL をツイートデータから削除した.

4 ツイートのベクトル化

SVM によるツイート推定においては単語表現に PN Table [5]を利用した. PN Table とは高村らの研究[6]をもとに各単語の感情極性値を-1 から+1 の範囲の実数値で表したものである. -1 に近い値ほど否定的な意味をもち, +1 に近い値ほど肯定的な意味を持つ単語である.

一方，LSTM によるツイート推定においては単語の分散表現が必要である．分散表現の獲得に深層学習のライブラリの keras[7]のエンベディングレイヤー[8]と日本語 Wikipedia エンティティベクトル[9]を利用した．すなわち，作成したコーパス中の学習用データでエンベディングレイヤーにより求めたベクトルと，日本語 Wikipedia エンティティベクトル（これは日本語 Wikipedia データを学習して得られたもの）を用いた．

4.1 SVM のためのベクトル化

SVM のためのベクトル化は，辞書の作成と辞書を用いた特徴ベクトルの作成からなる．辞書は以下の手順で作成する．

1. Mecab を用いてツイート（テストデータを除く）を形態素解析する．
2. 名詞（固有名詞，サ変接続，一般）または形容詞を抽出する．
3. 名詞が連続している場合，それら単独の名詞に加え，前後の名詞を結合したものも新たな名詞とする．
4. 抽出した名詞と形容詞をそれぞれ出現頻度の多い順に並べ，辞書とする．
   なお，名詞と形容詞の辞書を別々とする．

名詞はツイートの特徴を表す単語が多いため採用した．また，本研究は感情が関係しているため感情を表す形容詞も採用した．手順 3 は，連続して意味を持つ単語（例：日本会議，安倍晋三）に対応するために行った．一方，特徴ベクトルは以下のように作成する．まず，上記で作成された辞書内の単語を特徴ベクトルの要素とする．次に，その個々の要素の値は，以下の二つの方法で求める．

A. 単語の出現回数（以降，この方法を「**単語の出現回数**」と呼ぶ）
B. 単語を修飾する周辺の単語の感情極性値を足し合わせた感情値（以降，この方法を「**単語への感情値**」と呼ぶ）

「単語の出現回数」の場合，名詞からトップ N 以内の単語，形容詞からトップ M 以内の単語を抽出し，N と M の和は固定で 200 と設定し，N と M の比率を変更して特徴ベクトルを作成し複数のパターンで実験を行う．

「単語への感情値」の場合は N=200（M=0）とした．名詞の中には安倍晋三，ネトウヨ，パヨクなどが含んでおり，そのような単語に対する感情が推定に有効だと考える．また，単語への感情値の算出はその単語と係り受けの関係にある単語の感情値を用いるため，（M=0 であっても）結果的には名詞のみならず他の品詞の単語も考慮されることになっている．「単語への感情値」は以下の手順で求める．

1. ツイートを，改行，句点，空白で文に分割する．分割された各文に対し，以下の処理を行う．
   1.1 Cabocha を用いて形態素・構文解析を行う．
   1.2 各名詞についてその係り先の単語を抽出し，感情極性値を合計する．この手順を新たな係り先が見つかる間繰り返す．
   1.3 手順 1.2 で求めた各名詞の感情極性値の平均を算出する．
2. 手順 1.3 で算出した値（各文の各名詞の感情極性値の平均）の和を計算し，ツイートの感情値とする．

4.2 LSTM のためのベクトル化

LSTM への入力は単語の並びを考慮したものである．その入力（ここでツイートベクトルと呼ぶ）の作成手順を下記に述べる．

1. Mecab を用いてツイート（テストデータを除く）を形態素解析する．
2. 全異なり単語に対して番号を割り振り，各ツイートを単語の出現順に単語番号を要素としたベクトルに変換する．ベクトルの次元は1つのツイートに出現した単語の数となる．
3. 手順 2 のベクトルに対し，一番単語を多く使用しているツイートの単語数に合わせてゼロパディングを行う．ベクトルの次元は 101 まで統一される．
4. 手順 2，3 で作成されたベクトルの各要素（単語番号）に対し，単語の分散表現を代入しツイートベクトルを得る．ツイートベクトルの次元は 101×200 で，200 は単語の分散表現の次元である．ただし，単語の分散表現は以下の二種類を用いる．
   A. 学習データで求めた分散表現を用いる（以降，この方法を「**学習データ**」と呼ぶ）
   B. 日本語 Wikipedia データで求めた分散表現を用いる（以降，この方法を「**Wikipedia データ**」と呼ぶ）

5　ベースライン

ベースラインでは，ツイートから「自民党」と係り受けの関係にある単語の感情極性値を足し合わせた感情値からクラス推定を行う．具体的な推定方法は以下の通りである．まず，1つのツイートに対し，4.1 節の「単語への感情値」と同様の方法で「自民党」の感情値を求める．次に，「肯定的」なツイートを評価値 1，「批判的」なツイートを評価値-1，「言及していない」ツイートを評価値 0 として，算出された感情値とその三種の評価値との差の絶対値を求め，求めた値が一番小さい値のクラスを推定結果とする．

6　実験

自民党を含むツイートを，自民党に対して「肯定的」「批判的」「言及していない」のいずれかを，ベースラインと SVM と LSTM を用いて推定を行った．SVM は機械学習ライブラリ scikit-learn[10]の SVM(RBF)を利用する．LSTM は深層学習ライブラリ keras を利用した．

6.1　データ

作成されたコーパスの 2000 ツイートを学習データ，検証データ，テストデータの3つに分割する．学習データと検証データ:テストデータが 9:1，学習データ:検証データが 9:1 と設定した．すなわち，学習データが 1620 ツイート，検証データが 180 ツイート，テストデータが 200 ツイートであった．学習データと検証データ内の三つのクラスのデータの比率は「肯定的」「批判的」「言及していない」が 1:1:2 であった．

6.2　ハイパーパラメータの最適化

SVM(RBF)と LSTM の最適なハイパーパラメータはグリッドサーチを行い決定した．ハイパーパラメータの組み合わせについて，SVM(RBF)は γ と C がそれぞれ $10^{-4} - 10^{4}$ 間に対数（基底 10）スケール 15 分割で 225 通り，LSTM は中間層のユニット数が{3, 30, 50, 70 ,10}，学習回数（epoch）が{10, 20, 30, 40, 50}，バッチサイズが{1, 32, 128, 256}，出力層の活性化関数が{ Softmax, Sigmoid }で 200 通りであった．

6.3　実験結果と考察

6.3.1　SVM

特徴ベクトルの構成に「単語の出現回数」を使う場合においては，N と M の和を 200 に，N と M の比率をそれぞれ 1:0, 3:1, 1:1, 1:3, 0:1 に設定した．「単語への感情値」を使う場合においては名詞の個数 N を 200 個（M=0）に設定した．ハイパーパラメータセットを検証誤差の小さい順に並べ，上位 N 個を用いた場合のテストデータに対する平均精度を図 1 に示す．

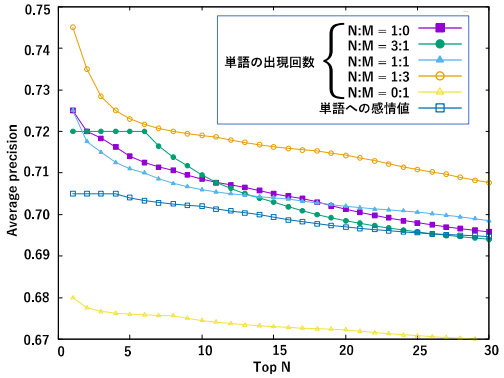

図1：SVM によるツイート推定の平均精度

図1より「単語の出現回数」を使う場合において，名詞と形容詞の比率 1:3 が最も平均精度が高かった．このことから「単語の出現回数」の場合，特徴ベクトルの構成に形容詞が重要であることがわかる．形容詞は感情を表すためツイート推定に有用であった．一方，0:1 は平均精度が全体的に低かったことから形容詞のみで特徴ベクトルを作成すると平均精度が下がることがわかる．このことは，特徴ベクトルには名詞と形容詞の共起性が重要であると考える．また，形容詞の使用数が少ない，または使われてないツイートの場合，特徴ベクトルが疎ベクトルとなるためそのようなツイートに対して推定出来ていなかった．さらに形容詞を少なめに使う 1:1 は 1:3 より精度が劣っている．これは形容詞の出現頻度上位のみで特徴ベクトルを作成すると肯定的なツイートの推定が行えていなかったためである．実際，政治に関するツイートにはネガティブな形容詞を使用したツイートの方が多かった．そのため形容詞の出現頻度上位は「悪い」「酷い」「おかしい」などのネガティブな単語が占めている．一方，中位（101 位-150 位）の名詞にはポジティブな意味を持つ単語が多く，中位を使用している 1:3 では肯定的なツイートも推定することが出来た．

次いで「単語の出現回数」と「単語への感情値」を比較する．「単語への感情値」は N=200（M=0）に設定しているため，同条件の 1:0 と比較すると，「単語への感情値」の平均精度の方が高かったことがわかる．「単語への感情値」の方が係り受け関係，そして単語の感情極性値を利用しているためよい結果が得られると考えていた．しかしツイートの中には「#」の後に文章または単語付与し，空白または改行を繰り返したものが多くある（例:「#自民党　#投票に行こう」）．このようなツイートに対し

ては名詞から係り先がない場合が多く感情値を求めることが出来ず，疎ベクトルとなることから推定が行えなかった．「単語への感情値」はこのようなツイート場合の特徴ベクトルを改善する必要がある．

6.3.2 LSTM

LSTM では「学習データ」を用いてベクトルを構成する手法と「Wikipedia データ」を用いてベクトルを構成する手法の精度の比較を行った．ハイパーパラメータセットを検証誤差の小さい順に並べ，上位 N 個を用いた場合のテストデータに対する平均精度を図2に示す．

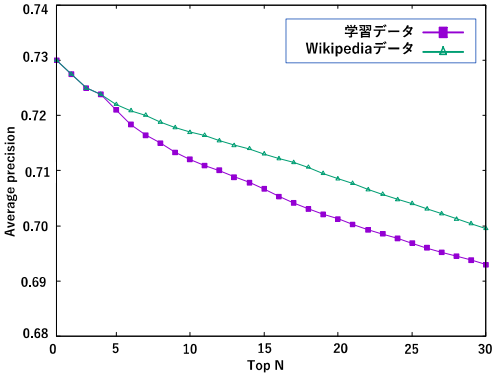

図2:LSTM によるツイート推定の平均精度

図2より，ベクトルの作成に手法「Wikipedia データ」を用いた方が，全体的に平均精度が高かったことが見て取れる．これは「Wikipedia データ」のほうが「学習データ」のより分散表現の学習における学習データの量が多かったことに起因するものと考える．

6.3.3 各手法の比較

表 1 に各手法の推定精度を示す．機械学習においては検証誤差が最小のハイパーパラメータセットを用いた時の精度である． また，SVM（単語の出現回数）の精度は一番精度の良い 1:3 のものを記載している．

表 1：各手法によるツイート推定の精度

ベースライン	SVM(単語の出現回数)	SVM（単語への感情値）	LSTM(学習データ)	LSTM (Wikipedia データ)
0.405	0.745	0.705	0.73	0.73

表1よりベースラインの精度が格段に低く SVM（単語の出現回数）のがもっとも高かったことがわかる．SVM（単語への感情値）の精度が SVM（単語の出現回数）のそれより低かった理由は 6.3.1 節で述べた通りである．一方，LSTM は時系列データにおける学習が可能なモデルであり語順などの文章としての情報を取り込んだ学習が行えるためもっともよい結果が得られると期待していた．しかし文章として不完全なものや，単語や人名を空白または改行で羅列したツイートに対して推定がうまく行えず高い精度が得られなかった．

7 おわりに

本稿では政党に対して支持・不支持のツイート推定を行うことを目的に，SVM と LSTM を用いた手法を提案した．学習データは，ツイートデータを取得し，それらに手動でタグ付けを行うことにより作成した．ツイートのベクトル化については複数の方法を考案した．提案手法の有効性を確認するために，複数の方法で構成したベクトルを用いた SVM と LSTM 間の比較実験，さらにはベースラインとして作成した手法との比較実験を行った．

実験の結果，提案手法によるツイート推定はある程度の精度を得ることが出来た．しかし本研究はユーザーの各ツイートの推定を行い，ユーザーの政党支持・不支持を判定して支持率を予測することを最終目標としている．正確な支持率の予測に，本稿の推定精度はまだ満足のいくものではなかった．今後の課題は，政党支持・不支持が明確なユーザーの特徴量の多いツイートを含めてデータ数を増加させることやベクトル化方法の改良を行うことでツイート推定精度の向上を図り，最終目標の支持率の予測を行うことである．

参考文献

[1] 世論調査, https://ja.wikipedia.org/wiki/世論調査

[2] 増井, 藤野, 山本. 2017. Twitter の多軸的感情と政党支持率との関係について. 言語処理学会第 23 回年次大会, pp.222–225.

[3] Yahoo!つぶやき分析, https://www.yahoohelp.jp/app/answers/detail/p/595/a_id/68519/~/「分析」について(リアルタイム検索)

[4] Twitter Developer, https//developer.twitter.com/

[5] Semantic Orientations of Words, http://www.lr.pi.titech.ac.jp/~takamura/pndic_en.html

[6] 高村, 乾, 奥村学. 2006. スピンモデルによる単語の感情極性抽出. 情報処理学会論文誌ジャーナル, Vol.47, No.02, pp.627–637.

[7] keras, https//keras.io

[8] keras emmbedding, https://keras.io/ja/layers/embeddings/

[9] 日本語エンティティベクトル, http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/

[10] scikit-learn, https//scikit-learn.org/stable/