Tweet解析によるユーザー埋め込み表現を用いた都道府県レベルでの位置推定

奥村 貴俊　　彌冨 仁

法政大学大学院　理工学研究科　応用情報工学専攻
{takatoshi.okumura.7e@stu., iyatomi@}hosei.ac.jp

概 要

スマートフォンなどの普及により,「位置」の持つ価値は日々高まっている．本報告では,Twitter 発言から, そのユーザの主たる位置を都道府県レベルで推定する手法を提案する．提案手法は各ユーザの発言から，潜在意味解析 (LSI) あるいは，FastText や Doc2Vec といった新しい手法でユーザの発言を代表する特徴の抽出を行った後，dorpout を備え過学習を抑えた 4 層 neural network でそのユーザの主たる位置を推定する．実験には 47 都道府県各 1000 ユーザからそれぞれ 500 tweet を取得（合計 2350 万 tweet）し，47 クラス識別を行った．その結果 LSI を特徴抽出に用いた場合が最も良好な結果を示し，平均正解率 62.9 %，都道府県ごとの正解率の平均 63.3%を達成した．

1　はじめに

コミュニケーション手段の多様化により，簡易かつ，より広範囲に情報発信が行えるSNSの需要は高まっている．SNSの中におけるTwitterの利用率は，2016年現在，Line(67.0%)，Facebook(32.3%) に次いで27.5%の第3位であり，これらのサービスの利用率は年々増加傾向である [1]．Twitterは140字以内の発言 (ツイート) を投稿する事ができるサービスであり，返信 (リプライ) や引用 (リツイート) などの機能や，GPS 情報を元に位置情報 (ジオタグ) を付与する事ができる．位置情報は，ある特定の地域に向けたプロモーションや，発信者の位置情報による様々な分析に用いられており，実際に，2011年3月11日に発生した東日本大震災では，情報の伝達手段としてTwitterが活用された [2]．その他，地震や台風などの災害検知 [3]，インフルエンザなどの感染症の流行分析 [4]，観光地推薦 [5] など多岐にわたる．位置情報は，様々な面で活用可能であるが，位置情報付き発言は，全体の発言に対して非常に少ない [6, 7]．そこで本研究では，各ユーザの発言を解析する事により，位置情報の取得を試みる．

Twitterの発言から位置情報を推定する試みとして，杉谷ら [8] は各tweetに対してジオコーディングを行った後，位置情報の付与が行えた発言に対し，文字数などの特徴を用いSVMによってユーザ位置の推定を行った．その結果，適合率約80%において再現率約43 %を達成したと報告している．しかし，この手法においては，ジオコーディングを行うことができた発言は全体の0.28%のみであり実用性には大きな課題が残る．また森國ら [9] は，単語のフィルタリング手法として，特定のエリアで出現する単語を高く評価する“ AF フィルタ ”を提案し，正解距離を30km以内とした時の適合率と再現率は，約79%と約54%であったと報告している．これらの手法は，発言単位で場所の推定を行っており，1発言が140文字とまでに制限され，短いためこれ以上の精度向上は難しいと推察される．高橋ら [10] は,「雨」のような局所的な単語を用い，位置情報が既知のユーザとの発言同期を利用し，約41 %の精度で未知なユーザの位置推定を行う事ができたと報告している．この報告では,「雨」以外の単語を用いる事で更なる精度向上が見込まれると報告している．しかしながらこの手法は，発言同期を行うために多量かつ長期間にわたるジオタグ付き発言データを取得し保持しておく必要がある．また，得られた精度も改善の余地が残されている．

Tweetごとに正確な発信位置推定が可能であれば利用範囲は極めて広いが，限られた文字列情報のみから正しい位置推定を行うことは不可能である．そのため，本研究では，実用的な応用を想定し，ユーザごとに，主たる活動場所の推定を目的とする手法を提案する．

2　提案手法

提案手法は (1) 発言取得部，(2) 特徴抽出部，(3) 識別部の3要素から構成される．(1) 発言取得部で対象となるユーザのtweet群を抽出し，(2) 特徴抽出部で，

tweet 群から各ユーザの位置推定のための特徴ベクトルに変換を行い，(3) 識別部で，それらを 47 都道府県のいずれかのクラスに分類を行う．以下詳細を示す．

2.1 発言取得部

Twitter の発言取得には，まず，Twitter の StreamAPI を用い，ランダムに各ユーザのプロフィールを取得した．本実験では，推定されたユーザ位置（都道府県名）の妥当性を評価するために，gold standard が必要になる．そこで，Twitter のユーザプロフィールの 1 つである場所（Location）フィールドに都道府県名が 1 つのみ記載されているユーザを本研究での解析対象とし，記載された都道府県を gold standard とした．また，発言元アプリケーションを参照し，BOT と考えられるユーザは除外した上で，500 発言以上行っているユーザを選定した．こうした条件を満たしたユーザを各 47 都道府県から 1,000 ユーザずつになるまで選択し，Twitter の RestAPI を用い各 500 発言ずつ，計 23,500,000 発言を取得した．これらの 47,000 ユーザ分の発言の内 9 割を学習用，残り 1 割を評価用のデータとした．また，取得した各発言に対し共通の前処理として，MeCab [11] による形態素解析を行った後，ノイズ処理を目的として，固有名詞のみ抽出した．

2.2 特徴抽出部

先述した形態素解析後のこの後の単語の頻度を元にしたベクトルである bag of words（BoW）を用いた場合，語彙数は 518,565 となり，各ユーザの特徴を表すベクタは非常に高次元かつスパースになる．このままでは過学習状態となり，後の識別処理が適切に行えない．そこで特徴抽出部では，トピックモデルとして以前から使われている，特異値分解と等価の潜在意味解析 (latent semantic indexing: LSI) [12] や近年自然言語処理分野で優れた成果が報告されている FastText [13] や Doc2Vec [14] を用いて次元数を削減，あるいは異なる低次元表現の獲得を行いユーザごとの特徴を抽出した．

2.2.1 LSI による特徴抽出

前処理済みの単語群から LSI のベクトルに変換するために，まず BoW から TFIDF ベクトルに変換を行った．その後 TFIDF ベクトルを特異値分解により LSI ベクトルに変換を行った．本実験では，次元圧縮後の累積寄与率調査に基づき LSI ベクトルの次元数として 1,200 とした．

2.2.2 FastText による特徴抽出

FastText [13] は，単語を，単語間の意味の距離を考慮した任意の次元数 D_F のベクトルに変換できる手法の 1 つであり，以前に提案され広く使われている Word2Vec [15] と比べ，高速に学習が行なうことができ，Word2Vec では，各単語を学習用いていたのに対し，FastText では，各単語に加え，文字 n-gram を学習に用いることで，単語の表記ゆれや活用形を考慮したベクトル表現を獲得することができる．本実験では，wikipedia の全文を用い予め FastText で学習させた学習済みモデルを用いた．FastText を用いて **2.1** で得られた疎行列の各要素をベクトルに変換した場合，{語彙数× FastText で単語を表現する次元数 D_F} の行列となる．後の MLP による学習，識別処理のため，この行列を語彙方向に平均化処理を行い，D_F 次元のベクトルに変換する処理を行った．また，予備実験の結果 D_F=300 とした．

2.2.3 Doc2Vec による特徴抽出

Doc2Vec [14] は Word2Vec [15] を文章単位に拡張したモデルであり，文章を任意 D_D 次元のベクトルに変換することができる．FastText のモデル同様，Wikipedia の本文を用い学習させた Doc2Vec の学習済みモデルを用いた．予備実験の結果 D_D=400 とした．

2.3 識別部

識別部では，2 層の中間層を持つ多層ニューラルネットワーク（MLP）により，ベクトル化されたユーザ情報から，47 都道府県の 47 クラス分類を行う．以下の表 1 に，MLP の構成を示す．各ニューロンの活性化関数には rectified linear unit（ReLU）を用い，MLP の 2 つの中間層の間，ならびに中間層と出力層の間は，汎化性能を向上させるため dropout を導入し，最小化する誤差関数には L2 ノルム正則化を導入した．

3 結果

各特徴抽出手法に応じたモデルについて，都道府県 47 クラス分類を行い，最終的な出力が一致した場

表 1: 多層ニューラルネットワークの構造

	LSI	FastText	Doc2Vec
入力層	1200	300	400
中間層 1	100	100	80
Dropout	0.5	0.5	0.5
中間層 2	40	50	40
Dropout	0.3	0.2	0.5
出力層	47		

合のみ正解とした．図 1,2,3 に特徴抽出手法に LSI, FastText，DoctVec をそれぞれ用いた場合の都道府県 47 クラスの識別問題に対する confusion matrix を示した．なお各図は，可視化のために都道府県ごとに [0-1] に正規化したものである．ここで，図中の ID と都道府県名の対応を表 2 にまとめた．また識別結果のまとめを表 3 に示す．表 3 において，Total Accuracy は全ユーザに対する正解ユーザの数を表し，Average Accuracy は，都道府県ごとの正解率の平均を表す．また，図 1 において正解率が低い関東近辺の識別割合の上位 3 件を表 4 に示す．

4 考察

今回の実験において，情報表現に古典的な特異値分解に基づく LSI が最も良好な結果を示した．47 クラス分類において，全体の正解率，また都道府県ごとの正解率の平均もともに約 63%を実現していることから，東京などの大都市のみを出力するといったこともなく，限られた情報のみから良好な結果が得られたものと考えられる．

FastText によるモデルの精度が芳しくなかった理由として，文書ベクトルを作成する際，各単語から生成されるベクトルを単純に平均化した値を用いた事による影響があると考える．こうした単語ごとの符号化モデルを使う場合には検討が必要である．

Doc2Vec のモデルでは，文書全体をベクトル化できるため，良好な結果が期待されたが，識別精度は低かった．これは Doc2Vec のモデルの学習に用いた Wikipedia コーパスの内容と，Twitter の文書に無視できない差があり，都道府県ごとの特徴差が取得できていなかったのではないかと考える．

LSI では，文書の順序などの情報が失われてしまい，この点では Doc2Vec より不利となるがベクトル生成自体に Twitter の文書を用いており，ある程度の都道府県ごとのトピックが取得する事ができたのではないかと考える．

関東近辺の正解率が低い理由として，関東近辺では通学や通勤などで近隣都県に移動するユーザが多かった点と，関東近辺で出現するトピックはその他の地域においても出現し，関東近辺固有の特徴が取得できなかったと考える．

5 展望

今回の実験では，合計 2000 万以上の tweet を解析したが，前処理や，LSI 以外の特徴ベクトルの生成法はまだ改善の余地が残されている．また，いずれの方法においても，前処理として形態素解析を行う必要があり，特に表記ゆれの激しい Twitter のテキストでは，形態素解析の精度の影響もあるため，今後形態素解析を用いず単語レベルで解析を行う手法を適用して，比較検討を行いたい．

参考文献

[1] 総務省, “平成 29 年版情報通信白書,” 2018.

[2] 由. 吉次, “東日本大震災に見る大災害時のソーシャルメディアの役割:ツイッターを中心に,” **放送研究と調査**, vol. 61, no. 7, pp. 16–23, 2011.

[3] T. Sakaki, M. Okazaki, and Y. Matsuo, “Earthquake shakes twitter users: Real-time event detection by social sensors,” *Proceedings of the 19th International Conference on World Wide Web*, pp. 851–860, 2010.

[4] M. J. Paul and M. Dredze, “You are what you tweet: Analyzing twitter for public health.” *Icwsm*, vol. 20, pp. 265–272, 2011.

[5] 中嶋勇人, 新妻弘崇, and 太田学, “位置情報付きツイートを利用した観光ルート推薦,” **研究報告データベースシステム（*DBS*）**, vol. 2013, no. 28, pp. 1–6, 2013.

[6] Z. Cheng, J. Caverlee, and K. Lee, “You are where you tweet: A content-based approach to geo-locating twitter users,” *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pp. 759–768, 2010.

[7] 橋本康弘 and 岡瑞起, “都市におけるジオタグ付きツイートの統計 (特集人と環境に見る高次元データフローの生成と解析),” **人工知能学会誌** = *Journal of Japanese Society for Artificial Intelligence*, vol. 27, no. 4, pp. 424–431, 2012.

[8] 杉谷卓哉, 白川真澄, 原隆浩, and 西尾章治郎, “教師あり機械学習を用いたツイート投稿時のユーザ位置推定手法,” **研究報告データベースシステム（*DBS*）**, vol. 2013, no. 26, pp. 1–8, 2013.

[9] 森國泰平, 吉田光男, 岡部正幸, and 梅村恭司, “ツイート投稿位置推定のための単語フィルタリング手法,” **情報処理学会論文誌データベース（*TOD*）**, vol. 8, no. 4, pp. 16–26, 2015.

[10] 高橋哲朗, “発言同期を用いたマイクロブログ著者の位置推定,” **研究報告自然言語処理（*NL*）**, vol. 2013, no. 17, pp. 1–7, 2013.

[11] T. KUDO, “Applying conditional random fields to japanese morphological analysis,” *Proc. of EMNLP-2004*, pp. 230–237, 2004.

表 2: 都道府県 ID と対応する都道府県名

都道府県 ID	0	1	2	3	4	5	6	7	8	9	10	11
都道府県名	北海道	青森県	岩手県	宮城県	秋田県	山形県	福島県	茨城県	栃木県	群馬県	埼玉県	千葉県
都道府県 ID	12	13	14	15	16	17	18	19	20	21	22	23
都道府県名	東京都	神奈川県	新潟県	富山県	石川県	福井県	山梨県	長野県	岐阜県	静岡県	愛知県	三重県
都道府県 ID	24	25	26	27	28	29	30	31	32	33	34	35
都道府県名	滋賀県	京都府	大阪府	兵庫県	奈良県	和歌山県	鳥取県	島根県	岡山県	広島県	山口県	徳島県
都道府県 ID	36	37	38	39	40	41	42	43	44	45	46	
都道府県名	香川県	愛媛県	高知県	福岡県	佐賀県	長崎県	熊本県	大分県	宮崎県	鹿児島県	沖縄県	

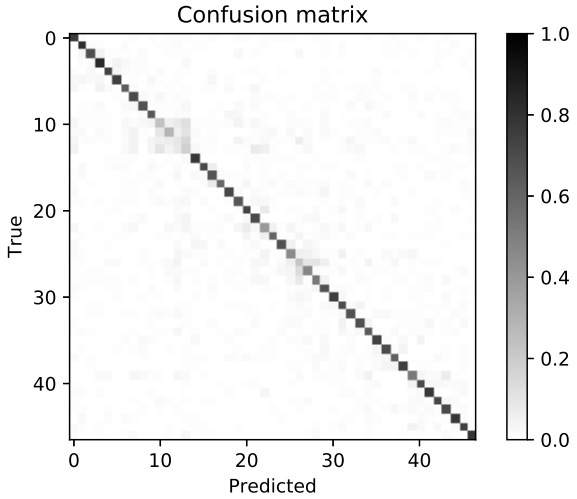

図 1: 特徴抽出に LSI を用いた際の判別結果

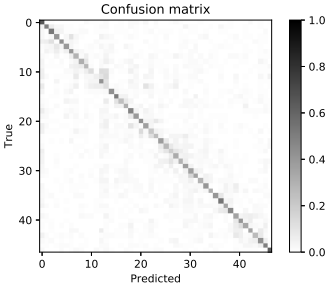

図 2: 特徴抽出に FastText を用いた際の判別結果

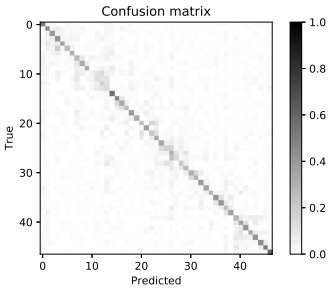

図 3: 特徴抽出に Doc2Vec を用いた際の判別結果

[12] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *CoRR arXiv:1607.04606*, 2016.

[14] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1188–1196, 2014.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems 26*, pp. 3111–3119, 2013.

表 3: 特徴表現手法の違いによる，都道府県の推定精度（%）の比較

	LSI	FastText	Doc2Vec
Total Accuracy	62.9	36.9	33.1
Average Accuracy	63.3	37.0	33.3

表 4: 埼玉，千葉，東京，神奈川における推定割合（%）

	1	2	3
埼玉 (ID:10)	埼玉 (23.9)	神奈川 (12.8)	群馬 (4.6)
千葉 (ID:11)	千葉 (29.3)	神奈川 (11.1)	埼玉 (10.1)
東京 (ID:12)	神奈川 (15.9)	東京 (9.7)	千葉 (7.1)
神奈川 (ID:13)	神奈川 (18.6)	埼玉 (10.8)	静岡 (7.8)