7P-05

Word Embeddingsによる特徴ベクトルを用いた文単位の評判分析

林　俊孝†　　藤田　ハミド†　　榑松　理樹†　　羽倉　淳†

岩手県立大学ソフトウェア情報学研究科†

１．はじめに

近年，テキストデータから特定の製品や組織などに対する評判を自動的に抽出して集約するための技術が注目を集めている．このような技術は評判分析や Sentiment Analysis などと呼ばれ，人工知能や自然言語処理のなどの分野において盛んに研究が行われている．評判分析においては，テキストを Positive と Negative に分類するのが一般的である．文や文書など，テキストの規模によって手法が変わるが，本論文では，文に対する評判分析を考える．

現在の評判分析は極性辞書を利用した手法[1)]が有力である．これに対し，本研究では，機械学習を用いて評判分析を行うことによって高い Accuracy を出すことを試みる．

自然言語を機械学習で扱うためには文書や単語を数値に変換しなければならないという問題がある．この点に対し，本研究では Word2vec[2)]を利用することで解決を試みる．Word2vec とは文書や単語をベクトル化して表現するニューラルネットワークである．入力がテキストコーパスで，出力が Word Embeddings となる．

２．提案手法

２．１　システムの流れ

本研究で提案するシステムの流れを図１に示す．

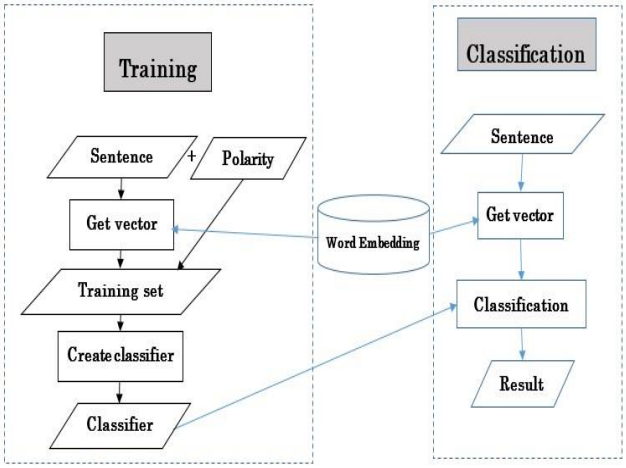

図１　システムの流れ

本提案手法は，大きく学習ステップと分類ステップに分かれる．両ステップともに，Word Embedding を用いて Sentence　Vector を取得する．この Vector が Sentence の特徴となる．学習ステップでは，Sentence Vector と Polarity を用いて，Polarity の分類器を作成する．分類ステップでは，作成した分類器を用い，新規文の Polarity を予測する．

２．２　Sentence Vector の取得の仕方

Sentence の Vector を求める手順を次に示す．

① 式(1)で示すように Sentence を単語に分ける．

$$\text{Sentence}="Word_1\ Word_2 \ldots Word_i \ldots Word_n" \quad (1)$$

② 文の i 番目の単語のベクトル $\overrightarrow{Word_i}$ を Word Embedding より求める．式(2)で示すように $\overrightarrow{Word_i}$ は m 次元のベクトルである．

$$\overrightarrow{Word_i} = (num_{i1}, num_{i2}, \ldots, num_{id}, \ldots, num_{im}) \quad (2)$$

同様に文中の全単語のベクトルを求める．

③ 文中の各単語に対し，②で求めた単語 Vector の平均$\overrightarrow{SA}$，分散$\overrightarrow{SV}$，幾何平均$\overrightarrow{SG}$を求める．それぞれ式(3)～(5)で示す．これらの値を要素とする Vector を作成し，これを Sentence Vector とする．

$$\overrightarrow{SA} = \frac{\sum_{i=1}^{n} \overrightarrow{Word_i}}{n} \quad (3)$$

$$\overrightarrow{SV} = (SV_1, SV_2, \ldots, SV_d, \ldots, SV_m) \quad (4)$$

$$SV_d = \frac{\sum_{i=1}^{n} (num_{id} - SA_d)^2}{n}$$

$$\overrightarrow{SG} = (SG_1, SG_2, \ldots, SG_d, \ldots, SG_m) \quad (5)$$

$$SG_d = \left(\sqrt[n]{\prod_{i=1}^{n} (1 + num_{id})}\right) - 1$$

２．３　分類器作成

２．２で取得した Sentence Vector を特徴，Polarity をラベルとして，分類器を作成する．本論文では教師あり学習の 1 つである xgboost[3)]を使って分類器を作成した．

Sentence-level Sentiment Analysis using feature vectors from word embedding

†HAYASHI TOSHITAKA †FUJITA HAMIDO
†KUREMATSU MASAKI †HAKURA JUN
†Graduate School of Software and Information Science, Iwate Prefectural University , g231p018@s.iwate-pu.ac.jp

３．評価実験

３．１　実験方法

提案手法の評価のため，以下の実験を行う．

データとして Twitter Sentiment Corpus Dataset[4]を利用する．

Dataset は，①から④の 4 つのサブセットに分割して利用する．各サブセットの Positive,Negative のバランスを表 1 で示す．

表 1　分割後のデータのバランス

ID	positive	negative	Total
①	233453	166547	400000
②	212330	187670	400000
③	177064	222936	400000
④	167331	211283	378614
合計	790178	788436	1578614

分割したデータ群のうち 3 つを使用して学習を行い，残りの 1 つを評価用データとし，Accuracy を求める．学習を行う際の xgboost のパラメータを表 2 に示す，また Vector 作成時に用いる Word Embeddings としては Google-News-Vector-Negative300.bin[5]を利用する．Accuracy については，式(6)を用いて求める．式で利用する記号の意味は表 3 に示す．

この作業を，評価用データを入れ替えて 4 回繰り返す．

表 2　xgboost のパラメータ

パラメータ	値
test_size	0.2
objective	binary:logistic
eval_metric	error
Eta(学習率)	0.1
max_depth	10

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)$$

表 3　式(6)中の記号

	Positive	Negative
予測が Positive	TP	FP
予測が Negative	FN	TN

３．２　実験結果と考察

実験結果を表 4 に示す．

表 4　実験結果

Train	Test	Accuracy
②③④	①	73.8
①③④	②	75.2
①②④	③	76.5
①②③	④	75.8
平均		75.3

Accuracy は平均 75.3%という結果となった．このことから Word Embedding は評判分析にも適用できると考える．

Accuracy を向上するためには，次にあげるような点を改善する必要がある．1 つ目としては，Sentence 中に Word Embedding に含まれない単語が出現した場合の対応である．今回は Word Embedding に含まれない単語は無視したが，文中の単語がすべて Word Embedding に含まれないという場合もあり，Accuracy を下げる大きな要因になったと考えられる．そのため，この点を改善する必要がある．2 つめとして，Sentence Vector を求める式についても改良の余地がある．今回提案した方法では文中の単語は全て同じ重みで扱っていたが，単語によって重みを変えることで精度が向上する可能性がある．3 つ目として，文法的なルールも加味する必要がある．たとえば not などの否定する表現が出現した場合，Sentence Vector に何らかの修正を加えなければ，分類に悪影響がでる可能性が高い．以上のような点を今後改善する必要がある．

４．おわりに

本論文では Word Embeddings を用いた Sentence-level Sentiment Analysis の手法について提案した．Sentence 中の Word Vector を Word Embeddings で求め，それらを用いて作成した Sentence Vector と Polarity から分類器を作成した．実験の結果，本提案手法の有用性を示した．

今後は，前処理の追加や，文法的なルールの対応などを行い，システムを改善していく．

参考文献

1) Orestes Appel, Francisco Chiclana, Jenny Carter, Hamido Fujita, A Hybrid Approach to the Sentiment Analysis Problem at the Sentence Level, Knowledge-Based Systems (2016), Volume 108, 15 September 2016, Pages 110-124.
2) T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean, Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems, 3111-3119
3) Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). ACM, New York, NY, USA, 785-794.
4) Twitter-Sentiment-Aanalysis-Corpus-Dataset(http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/)(閲覧日2018/01/08)
5) Google-News-vectors-negative300.bin (https://github.com/mmihaltz/word2vec-GoogleNews-vectors)(閲覧日2018/01/08)