文書分類に適したWord Embeddingの非線形変換法

Daniel Andrade　田村 晃裕　土田 正明
NEC 情報・ナレッジ研究所
{s-andrade@cj, a-tamura@ah, m-tsuchida@cq}.jp.nec.com

1　はじめに

近年，Word Embeddingは，品詞解析，同義文判定といった様々なNLPタスクで有効であることが示されている．WEとは，低次元なベクトルで，単語の語義や統語的な情報を表現しているものとされている．そのため，例えば，WEのユークリッド距離が近い単語は意味的にも類似している傾向にある．

一方で，WEは文書分類への応用では成功例が少なく，文書分類で効果を発揮させるには，文書分類に特化したWEを使う必要があることが報告されている[5, 8]．例えば，汎用的なWEの学習では，一般に語義が近い「やばい」と「危ない」のWEが近くなりやすいが，映画の評判では「やばい」の語義は「すばらしい」という意味で使われることが多い．そこで，映画の評判分析では，映画の評判文書とそれに付与されているpositive/negativeのラベル情報を使ってWEを変換することで,「やばい」と「すばらしい」のWEを近づける必要がある．

先行研究[5, 8]では，ラベル付き文書を学習データとして，個々のWEをパラメータとみなして文書分類タスクにそのWEを最適化する．したがって，先行研究は，ラベル付き学習データに存在しない単語のWEは学習できないという問題がある．

そこで，本論文では，文書分類タスクに合わせて，元のWEの空間の非線形変換を学習する方法を提案する．提案手法により得られた非線形変換は，ラベル付き文書に存在しない（元のWEを学習したラベルなし文書にのみ存在する）単語のWEにも適用・調整できるため，学習データが十分多くない場合でも有効と考えられる．

映画の評判データIMDB[6]のpositive（評判がいい）/negative（評判が悪い）の分類タスクによる評価実験で，学習データが中規模(1000件)の場合，[5]と比較して精度が1%以上向上することを確認した．

2　従来手法

[5]は，文書分類タスクに適したWEの学習方法として，既存のWE[1]を，分類ラベル付き文書の学習データを用いて調整する方法を提案した．各単語wに対応するベクトル（WE）eが独立に生成されると仮定すると，WEを調整するパラメータベクトルϕが与えられた際，文書ラベルcの条件付確率は以下のようなロジスティック回帰で決まる．

$$p(y = c|w) = \frac{1}{1 + exp(-\phi^T e)} . \quad (1)$$

単語ベクトルeとパラメータベクトルϕは以下の目的関数によって最適化する．

$$\underset{\phi, E'}{\operatorname{argmin}} - \sum_i \sum_t \log p(y = c_i|w_t) + \lambda||E' - E^{orig}|| .$$

ここで，iは文書のインデックス，tは各文書の単語のインデックスを表す．また，E'は新しいWEからなる行列，E^{orig}は元々の単語のWEからなる行列を表す．E'はパラメータであり，E^{orig}は固定されている．$||\cdot||$はフロベニウスノルムを表す．ハイパーパラメータλは，元のWEの情報をどの程度残すかを調整する．ここで，w_tは学習データに存在する単語のみとなるため，学習データにない単語は調整できないことが分かる．

[8]は，既存のWEを初期値とせずに，文書分類タスクに合わせたWEの学習方法を提案している．ただし，WEの学習には各文書のラベルが必要であるため，ラベル付き文書の量が十分でない場合，様々な文脈の情報を含むWEは得られないと考えられる．そこで，彼らはツイートにある絵文字によって文書に擬似ラベルを振ることで，大量の学習データを自動生成した．大量の学習データがあれば，出現しない単語は少なくなると考えられるが，このようなヒューリスティスクが使えない分類タスクでは，大量のラベル付き文書を用意することが困難という問題がある．

[1] 一般的な方法で学習したWE

3 提案手法

提案法では，以下の非線形変換により，既存の WE を分類タスク用 WE に変換する．

$$e' = \alpha \cdot \tanh(\frac{1}{\alpha} \cdot T \cdot e^{orig}), \quad (2)$$

元の WE を e^{orig}，変換後の WE を e' とする．$\alpha \in \mathbb{R}$ はスケーリング用の定数である．$T \in \mathbb{R}^{d \times d}$ は行列であり，WE を変換するためのパラメータで，学習データから学習する．関数 tanh は要素ごとに施すものとする．

また，文書ベクトルは，単語の WE を重み付き平均したベクトルを用いる．すなわち，文書 i のベクトル x_i を

$$x_i = \sum_{t=1}^{n_i} q_t \cdot e'_t, \quad (3)$$

とする．n_i は文書中の単語の数，q_t は単語の重み，e'_t は t 個目の単語の変換後 WE である．q_t は単語の idf を正規化した重みを使う．

$$q_t \propto \log \frac{D}{f_{w_t}}, \quad (4)$$

D は学習データの文書数で，f_{w_t} は単語 w_t の文書頻度である．

行列 T を学習するために，文書分類問題をロジスティック回帰で表現する．

$$p(y_i = c|x_i) = \frac{exp(x_i^T \phi_c + b_c)}{\sum_{k=1}^{n_c} exp(x_i^T \phi_k + b_k)}, \quad (5)$$

$\phi_k \in \mathbb{R}^d$ と $b_k \in \mathbb{R}$ は説明変数のパラメータで，n_c はクラスの数である．[2] 学習では，次の負の対数尤度を最小にするパラメータ T，ϕ_k，b_k を求める．

$$-\sum_{i=1}^{n} \log p(y_i = c|x_i) + \lambda_1 \cdot r_1(\phi) + \lambda_2 \cdot r_2(T), \quad (6)$$

r_1 と r_2 はそれぞれ w と T の正則関数であり，λ_1 と λ_2 はハイパーパラメータである．r_1 は通常の L2 正則化を用いる．r_2 は次節で説明する．

3.1 行列 T の正則化

式 (2) より，定数 α が十分高く（$\alpha \gg 1$），かつ，行列 T が単位行列であれば，$e' \approx e^{orig}$ となる．その理由は，tanh は入力の絶対値がゼロに近い場合にほぼ線形であり，$\tanh(\frac{1}{\alpha} \cdot e_t^{orig}) \approx \frac{1}{\alpha} \cdot e_t^{orig}$ となるためである．そこで，元の WE の空間の情報を残せるように，α を十分高く設定し，行列 T が単位行列 I に近くなるように正則化する．

$$r_2(T) = ||T - I||^2,$$

$||\cdot||$ はフロベニウスノルムを表す．

3.2 ハイパーパラメータの設定

定数 α は十分大きくするために，以下のように全ての元々の WE における最大値に設定する．

$$\alpha := max_e \, max_{l=1}^{d} e^{orig}(l),$$

e は全ての単語の WE の範囲であり，d は WE の次元数である．

ハイパーパラメータ λ_1 は以下の値に固定する．

$$\lambda_1 := \frac{1}{d \cdot n_c}.$$

前述した通り，ハイパーパラメータ λ_2 は e' がどれぐらい e^{orig} から離れるかを決める重要なパラメータであるため，従来方式 re-embedding[5] と同様に学習データの 2 割を使いチューニングする．

3.3 パラメータの学習

式 (2) より，式 (6) の最適化問題は，tanh の影響で非凸である．そこで，勾配降下法で局所最適解を求める．具体的には，AdaGrad[2] を利用して，マスター学習率を 1.0 に設定する．行列 T の初期値は単位行列にして，反復数は 1000 回にする．

4 実験

手法	1000 件	5000 件
提案手法	**0.830**	**0.848**
Re-embedding	0.817	**0.848**
Original	0.817	0.820

表 1: 文書分類の評価結果（学習データが 1000 件と 5000 件の場合）

実験は，映画の評判コーパス IMDB[6] を使い，文書が positive か negative かを分類する 2 値分類タスクで評価した．ベースライン方式として，元の WE

[2] 本論文では 2 値分類で実験しているため，n_c は 2 である．

手法	1000 件	5000 件
提案手法	**0.822**	**0.855**
Re-embedding	0.815	0.849
Original	0.817	0.820

表 2: 学習データにある単語のみを素性に使った文書分類の評価結果（学習データが 1000 件と 5000 件の場合）

(Original) と Re-embedding[5] と比較した．IMDB には，ラベルなし文書 5 万件とラベル付き文書 5 万件が含まれている．ラベル付き文書は，学習用と評価用にそれぞれ 2 万 5 千件ずつ分かれており，各文書には，positive/negative の 2 値ラベルが付与されている．実験では，文書分類の評価には，評価用文書の中から選んだ 1 万文書を使った．また，WE の調整と文書分類器の学習には，1 千件（または 5 千件）の学習用文書を用いた．

実験手順について述べる．まず，全ての文書に対して Senna[1] で単語分割と原形変換を行った．そして，IMDB にある学習用文書とラベルなし文書を合わせた 7 万 5 千件のコーパスに Word2Vec[7] をかけて，50 次元の WE を求めた．[3]

その後，WE から文書のベクトル表現を式 (3) のように求めた．「Re-embedding」方式の際には，式 (3) の e' は e^{orig} を Re-embedding で調整した WE であり，「Original」方式では e^{orig} そのものである．式 (3) の文書表現には，学習データにない単語の WE も利用できることに注意されたい．そのため，テストの際には，初めて出現している単語の文書頻度を 1 とする（式 (4) の d_{w_t} を参照）．最後に，求めた文書のベクトル表現により文書分類を行う．分類器の違いの影響をなくすために，全ての方式で，文書分類器は LIBLINEAR のロジスティック回帰 [3] を使い，パラメータ C は 5 分割交差検定で定めた．とりわけ，提案手法は行列 T を学習する際に式 (5) のとおり文書分類器も同時に学習できるが，文書分類の評価の際には LIBLINEAR のロジスティック回帰を利用した．

提案手法とベースライン方式の性能を表 1 に示す．文書分類性能の尺度は，再現率と適合率が一致している break-even-point を利用する．

まず，「Re-embedding」方式と「Original」方式の差は [5] で報告されたほど大きくないことが分かる．その理由は，[5] では，Original は別のコーパスから学習した WE であるが，本実験では，文書分類の対象となるコーパスから学習した WE を使用したからであると考えられる．

表 1 より，学習データが 1000 件の場合には，提案手法は従来方式より優れていることが分かる．これは，提案手法は学習データにない単語の調整も可能なためと考えられる．それを確かめるため，学習データにしかない単語に絞った評価も行った．結果を表 2 に示す．表 2 より，学習データが 1000 件の場合，提案手法の長所はそれほど発揮できないことが分かった．一方で，表 1 より，学習データが 5000 件の場合には，提案手法と従来手法は同等の性能である．これは，学習データが大規模になると，分類に重要な単語の大部分が学習データに含まれてしまうためと考えらえる．

4.1 極性語の分析

	1000 件	5000 件
内語	2545 (887/1658)	3939 (1244/2695)
外語	2154 (567/1587)	760 (210/550)

表 3: 極性辞書中の内語と外語の合計数（positive 数/negative 数）

学習データ **1000** 件				
	内語		外語	
手法	上位 10	上位 100	上位 10	上位 100
提案手法	**0.727**	**0.687**	**0.679**	**0.660**
Re-embedding	0.726	0.665	0.672	0.654
Original	0.726	0.665	0.672	0.654
学習データ **5000** 件				
	内語		外語	
手法	上位 10	上位 100	上位 10	上位 100
提案手法	**0.723**	**0.687**	0.638	**0.625**
Re-embedding	**0.723**	0.681	**0.639**	0.624
Original	0.721	0.678	**0.639**	0.624

表 4: 極性語による内部評価（上位 10/100 語が同じ極性を持つ割合）

WE 自体の性質を評価するために，[8] と同じように，極性語の類似度実験を行った．映画評判の 2 値分類によって調整した WE が極性をより捉えられるようになったと仮定して，以下の評価を行った．[4] の辞書の単語を，学習データにある単語「内語」と学習データにない単語「外語」に分けた．そして，以下の通り，[8] で提案された尺度を使い，同じ極性を持つ単語の WE が異なる極性を持つ単語より近くなるかどうかを測定する．

$$\text{上位 } K \text{ 精度} = \frac{\sum_{w \in S} \sum_{k=1}^{K} \beta(w, c_k)}{K \cdot |S|},$$

[3] 全体のコーパスにある頻度 5 以上の単語のみ用いる．Word2Vec の CBOW モデルを利用して，他のパラメータもディフォルトにした．

提案手法		
順位	単語	極性
類似語 c1	dejectedly（落胆した）	[-]
類似語 c2	jealously（嫉妬）	[-]
類似語 c3	irregular（不規則）	[-]
従来手法		
順位	単語	極性
類似語 c1	maturely（熟した）	[+]
類似語 c2	illogically（不合理的）	[-]
類似語 c3	enraptured（うっとりした）	[+]

表 5: 外語「erratically」（迷走的）との類似語（Original と Re-embedding は同じ結果であるため「従来手法」にまとめる）

ここで，S は極性辞書にある単語集合（「内語」または「外語」）である．$c_k, k = 1..K$ は単語 w と似ている上位 K 単語である．単語の類似度は単語と対応している WE のユークリッド距離によって測る．$\beta(a, b)$ は単語 a が単語 b と同じ極性を持つ場合に 1，そうでない場合に 0 となる関数である．結果を表 4 に示す．

学習データが 1000 件の際には，提案手法は主に「内語」でよりよく極性を反映した WE の学習できるとみえるが，実際には，表 5 の例でみるように，「外語」の場合でも単語の WE の近さが大きく改善できる場合がある．

一方，5000 件の際には，内語でも外語でも従来方式との差が少なくなった．

5 おわりに

本論文では，行列 T によって，既存の WE を非線形変換し，文書ラベルの情報を WE に組み入れることができる手法を提案した．提案手法は，従来方式に比べて，ラベル付き学習データにない単語の WE も調整し，文書分類に最適化することができる．映画の評判コーパスを用いた実験において，中規模 (1000 件) の学習データでは，従来方式より高い文書分類精度を実現できることを確認できた．さらに，極性辞書による分析で，極性の情報を組み込むこともできることを確認した．

外部分析（文書分類の評価）と内部分析（極性辞書による評価）より，提案手法は学習データにない単語だけではなく，学習データにある単語の WE も従来方式よりうまくラベル情報を反映させることがと分かった（例えば，表 4 の学習データ 1000 件，上位 100 の精度を参照）．これは，Re-embedding 方式と比較すると，非線形変換を行ったことが理由の一つに考えられる．この点は，今後，より深く分析する予定である．

参考文献

[1] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[2] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[3] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[4] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[5] Igor Labutov and Hod Lipson. Re-embedding words. In *ACL*, pages 489–493, 2013.

[6] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *International Conference on Learning Representations (ICLR)*, 2013.

[8] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565, 2014.