日本語単語ベクトルの精度向上のための前処理手法の検討

西村 陸 † 松本 忠博 †
† 岐阜大学大学院工学研究科

1 はじめに

自然言語処理の分野において，機械に単語の意味を理解させることは対話システムや機械翻訳，構文解析，人工知能に研究など様々なタスクで成果が期待できるため，現在盛んに研究されている．単語の意味を計算機で扱うための様々な手法が提案されている中，特に大きな評価を与えられているものの一つに単語の分散意味表現として単語ベクトルを構築する手法がある．

ベクトル空間モデルに基づいた単語の意味表現構築の手法には今日様々なものが提案されているが，そのほとんどの評価が英語を対象にしており，他の言語での有用性はほとんど実証されていない．本研究では日本語において精度の良い単語ベクトルを収得することを目標としているが，日本語と英語では字数種，語順，文法などが大きく異なっているため，既存の手法をそのまま日本語に用いただけでは十分な成果が得られない可能性がある．

そこで本研究では日本語コーパスを様々な手法で加工することにより，既存の単語の意味表現構築の手法を用いることでより精度の高い日本語単語ベクトルを取得することを目的とする．

2 単語ベクトル

2.1 単語ベクトルの概要

単語の意味を機械上で表現する手法には様々なものがあり，その中の一つとして，ベクトル空間モデルに基づいた手法がある．この手法は，テキストデータにおける単語の出現頻度や共起頻度等を基に単語を多次元空間上に配置する手法であり，単語をベクトルとして表現するものである．単語間の関係などがその単語に対応したベクトルの差異などに反映されており，主観的でなく統計的な意味表現を獲得することができる．ベクトルの生成方法の特徴から新たに作られた単語などにも即座に対応することができる．

2.2 日本語単語ベクトル

日本語文章は英語などと違い単語ごとに分かち書きされておらず，単語ベクトルを構築する際どこからどこまでが一単語であるか記す必要があるので単語ごとにスペースで区切らなければならない．本研究では形態素解析ソフト MeCab[1] を用いて単語ごとに分割する．

3 Word2vec

Word2vec[2] とは mikolov らによって提案された，ニューラルネットワークを用いて単語の分散表現を獲得する手法であり，正確には，その手法が実装されたオープンソースソフトウェアの名称である．この手法では学習を実現するためのネットワーク構造として，Continuous Bag-of-Word (CBOW) モデルと Skip-gram モデルの二つのモデルを提案している．図 1 は CBOW モデルのネットワーク構造を表している. 図が表すように CBOW モデルはある単語 w_t に対して周辺単語 $w_{t-k}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_k$ を入力とし,wt を出力とする (予測する) ニューラルネットワークである．入力と出力は $1-of-K$ 単語ベクトルとなっている. 中間層は隠れ変数となっており, 任意の個数のノードを設定することができる. また CBOW モデルでは計算の簡略化のために単語 w t の前後に出てくる単語の語順を無視しているため, 文法的な意味を加味することができない．図 2 は Skip-gram モデルのネットワーク構造を表している.Skip-gram モデルは, ある単語 wt を入力とし, 入力単語の周辺に出現する単語 $w_{t-k}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_k$ を出力することで周辺の単語を予測するモデルとなっている. Word2vec ではテキスト集合である学習データの文脈をもとに単語の関係性をベクトル化して扱うことを可能にする．

この 2 つのモデルを用いて学習を行うことで，生成された単語のベクトル情報には単語の前後の文脈情報が埋め込まれている．また Word2vec の特徴の一つとして単語ベクトルの演算が可能であり，意味構造まで

もがベクトル化されているので意味構造自体を定量的に扱うことができる．

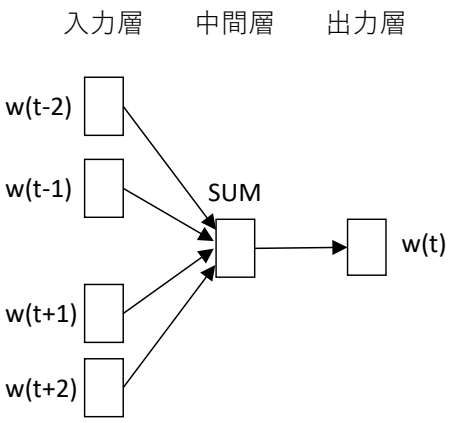

図 1: CBOW の概念図

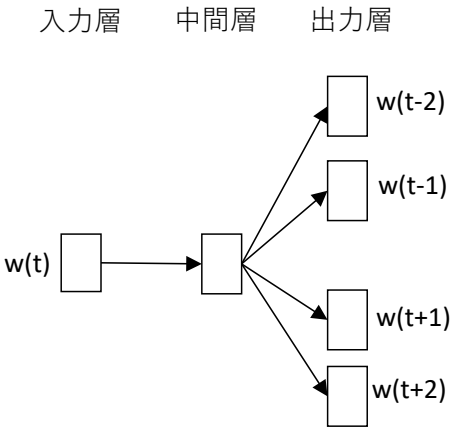

図 2: Skip-gram の概念図

4　日本語単語ベクトルの構築

4.1　データセット

本研究では，単語ベクトルの精度を図るための日本語テキストコーパスとして日本語 Wikipedia のテキストデータを使用した．使用したテキストファイルのテキストデータはおよそ 6GB であり，総単語数は約 11 億単語であった．

4.2　提案手法

本研究では日本語テキストコーパスを加工することで，単語ベクトルの精度を向上させることを目的としている．日本語の文章において単語ベクトル構築する際に不利となる要因を排除するため, いくつかの方法でコーパスを加工し，Word2vec を用いて単語ベクトルを生成する．word2vec のオプションはウィンドウサイズは最大 8 単語 (-windows 8) とし, ネガティブサンプリングのネガティブサンプルの個数は 25 個 (-negative 25) とし, 階層的ソフトマックスは使用しなかった (-hs 0). また, 中間層のノード数は 200 次元とした．提案する加工方法は以下の 6 通りである. 例を表 1 に示す.

- M1:助動詞を削除したコーパス
- M2:助詞を削除したコーパス
- M3:助動詞と助詞を削除したコーパス
- M4:数や記号を削除したコーパス
- M5:数とひらがな一文字を削除したコーパス
- M6:動詞と助動詞を基本形に変換したコーパス

表 1: 加工例

加工前	日本 の 首都 で ある 東京 都 に は 1362 万 人 が 住ん で いる
M1	日本 の 首都 東京 都 に は 1362 万 人 が 住ん で いる
M2	日本 首都 で ある 東京 都 1362 万 人 住ん いる
M3	日本 首都 東京 都 1362 万 人 住ん いる
M4	日本 の 首都 で ある 東京 都 に は 万 人 が 住ん で いる
M5	日本 首都 ある 東京 都 万 人 住ん いる
M6	日本 の 首都 だ ある 東京 都 に は 1362 万 人 が 住む で いる

4.3　加工理由

日本語において助動詞や助詞は付属語であり，主に自立語同士の関係性を表したり動詞に意味を付加する働きをするが，それ自体が数や記号と同じように定まった意味を持たない．そのような単語を取り除くことで自立語間の距離を縮め, より密なベクトルを構築することできる．また，ひらがな一文字の単語などは同じ単語であっても様々な用途で使われる場合が多く，一つの単語を一つの意味ベクトルでしか表すことのできない単語ベクトルにおいてそのような単語は曖昧なベクトルとして表現され，他の単語ベクトル構築の邪魔になる可能性が高い．

また、日本語において動詞や助動詞の活用形の多さも単語ベクトルの精度を下げている要因の一つであり，

すべての活用形を基本形に統一することで単語の出現頻度を上げ，より正確な意味ベクトルを構築が可能になると考える．

5 実験

5.1 精度検証方法

日本語のベクトルの精度を検証する方法として，ある関係性を持った単語のペアに対し，別の単語に対し同じ関係性を持つ単語を類推する問題を Word2vec に実装されているベクトル演算を用いて解き，その正答数によってベクトルの精度を検証する．問題には国-首都の関係性，ある単語と対義語になる単語の関係性の二つを用いる．問題数はそれぞれ 178 個と 979 個用意した．表 2 に作成した問題の例を複数示す．

表 2: 精度検証のための問題例

単語の関係性	基になるベクトル	問題例	正答	問題数
国-首都	日本-東京	フランス-?	パリ	178
		イギリス-?	ロンドン	
		中国-?	北京	
対義語	影-光	欠点-?	利点	979
		悪意-?	好意	
		短所-?	長所	

5.2 実験結果と考察

表 3: 実験結果

単語の関係性	国-首都		対義語		TOTAL	
問題数	178		979		1157	
	正答数	正答率	正答数	正答率	正答数	正答率
加工前	11	0.0617	185	0.1889	196	0.1694
M1	13	0.0730	194	0.1981	207	0.1789
M2	16	0.0898	185	0.1889	201	0.1737
M3	28	0.1573	187	0.1910	215	0.1858
M4	13	0.0730	192	0.1961	205	0.1771
M5	30	0.1685	189	0.1930	219	0.1892
M6	14	0.0786	188	0.1920	202	0.1745

表 3 の実験の結果から，何も手を加えていない Wikipedia コーパスに対し，加工後のすべてのコーパスで問題の正答率が向上するというおおよそ理想的な結果を確認できた．特に顕著な精度向上が見られた国と首都の関係性の問題について考察する．

表 4: 正答例

M3		加工前	
アイスランド- レイキャヴィーク	0.473		
アルゼンチン - ブエノスアイレス	0.537		
イラン - テヘラン	0.507	イラン - テヘラン	0.516
ウクライナ - キエフ	0.519		
エリトリア - アスマラ	0.467		
キプロス - ニコシア	0.505		
グルジア- トビリシ	0.520		
クロアチア - ザグレブ	0.521		
ジャマイカ- キングストン	0.491	ジャマイカ - キングストン	0.477
シリア- ダマスカス	0.498		
スリナム - パラマリボ	0.450		
ソマリア- モガディシュ	0.464		
大韓民国- ソウル	0.516	大韓民国 - ソウル	0.519
タイ - バンコク	0.583	タイ- バンコク	0.525
ドイツ- ベルリン	0.452		
トンガ - ヌクアロファ	0.465		
ハイチ- ポルトープランス	0.477		
パラグアイ - アスンシオン	0.539		
ハンガリー- ブダペスト	0.5217		
バングラデシュ - ダッカ	0.498	バングラデシュ - ダッカ	0.486
フランス - パリ	0.494	フランス- パリ	0.478
ベトナム - ハノイ	0.519		
		ベリーズ- ベルモパン	0.421
ベルギー- ブリュッセル	0.500	ベルギー- ブリュッセル	0.469
ミャンマー- ヤンゴン	0.556	ミャンマー- ヤンゴン	0.514
メキシコ- メキシコシティ	0.518	メキシコ- メキシコシティ	0.496
ルーマニア - ブカレスト	0.545		
レバノン - ベイルート	0.501	レバノン- ベイルート	0.472

全体として，アメリカのニューヨークのように国に対して首都以外に世界的有名な都市が存在する国の問題正答率が低くなる傾向にあった．原因としては首都に対してベクトル空間上の近い位置に多くの都市名の単語が存在しているからだと考えられる．逆に首都以外にあまり有名な都市が存在しない国は問題の正答率が高くなる傾向が見られた．

実験結果から国と首都の関係を予測する問題の正答率には加工方法ごとに大きな差が見られた．基本的に手の加えられていないコーパスが正答している問題は他の加工後のコーパスでも正答しており，どちらのコーパスでも正答している問題においても加工後のほうがより近い位置に正答のベクトルが存在しているという結果から，精度が向上されていることがわかる．最も正答率に差が出た M3 と加工前の国と首都の問題の正答と距離の一部を表 4 に示す．

また, 対義語を用いた問題では国と首都を用いた問題よりも正答率に大きな差が見られなかったが, 手のくわえられていないコーパスを手のくわえられたコーパスすべてが正答率を上回る結果が示された. しかし国と首都の関係での問題のように顕著な差は現れなかった．原因としては. 国や首都などの単語は主語や目的語として使われる場合が多いことや. 今回の実験では正答を一つに絞っているのに対し, 対義語の正答とな

る単語と似た意味の単語が多数存在することが理由であると考えられる.

6 おわりに

本研究では日本語コーパスから特定の品詞を取り除くなどの処理をすることでより精度の高い単語ベクトルを構築することに成功した．しかし，単語ベクトルはコーパスの性質によって大きく変化するため他のコーパスにおいても同じ方法で精度が向上するとは断言できない．そのため Twitter コーパスなど他のコーパスでも同じように精度が向上するか確認していく必要がある．また, 単語ベクトルの構築ために本研究では Word2vec を用いたが他のソフトウェアでも同じような結果を出せるかについても検証していく必要がある.

参考文献

[1] MeCab: Yet Another Part-of-Speech and Morphological Analyzer (http://mecab.sourceforge.net/)

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Proceedings of Workshop at ICLR, 2013.