単語のベクトル表現による 文脈に応じた単語の同義語拡張

有賀 竣哉　　鶴岡 慶雅
東京大学 工学部電子情報工学科
{ariga, tsuruoka}@logos.t.u-tokyo.ac.jp

1 はじめに

Benjio らによるニューラルネットワーク言語モデル (Neural Network Language Model, NNLM) [1] の出現以後、単語のベクトル表現は様々な学習方法が提案され、種々のタスクに応用されている。情報検索の分野においても、近年単語のベクトル表現によるアプローチが出現している [3, 6] が、情報検索の課題の 1 つに、文脈によって意味が変わる多義語の存在がある。図 1 のように単語が出現する文脈に応じて同義語を付与できれば、付与した同義語を用いて、単純な文字列マッチを超えた高度な検索を行うことが可能になる。

そこで本稿では、文脈から単語の同義表現を予測して付与することで、単語の意味を柔軟に解釈可能にすることを目的とする。そのため、2013 年 Mikolov らが公開した単語のベクトル表現の学習ツールである word2vec で用いられている Continuous Bag-of-Words (CBOW) [4] モデルが周辺単語から中心単語を予測する構造であることに着目した。CBOW モデルは前後の文脈を用いて中心単語を予測するモデルだが、“Bag-of-Words” の名が示す通り語順が無視されているため、実際に中心単語を予測し、同義語を付与するには十分でない。その上、word2vec で学習されたベクトルは、ベクトル同士の類似度の測定に使われることはあっても、周囲の単語から中心の単語を予測するタスクには使われていない。

本稿では CBOW に語順情報を付加した新たなモデルを提案する。実験で対数尤度を測定した結果として、提案モデルによって単語の予測精度が向上していることが示された。また、ある単語が異なる文脈で現れた時に、それぞれの文脈に即した同義語表現の付与が可能になった。提案モデルは単純な構造で高速に学習が可能なため、今後情報検索のみならず多方面に応用可能であると期待される。

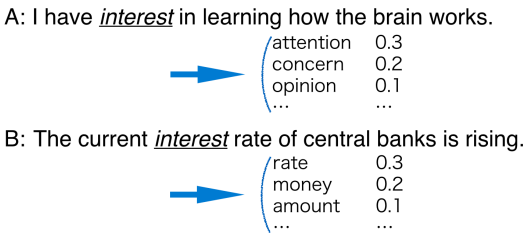


図 1. 文脈に応じた多義語の同義表現付与の例

2 CBOW モデル

CBOW は NNLM をベースにしているモデルである。NNLM では文脈として単語 w_t よりも前の単語のみを用いて次の単語を予測するが、CBOW は図 2(a) のように、前後の単語を文脈として用いて中心単語を予測する。中間層 H は式 (1) のように単語ベクトル V の足し合わせによって得る。

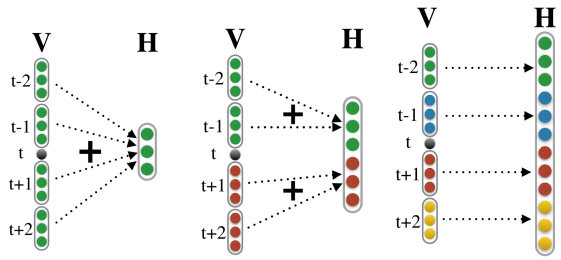
$$\boldsymbol{H_{CBOW}}_t = \sum_{-R \le j \le R, j \neq 0} V(w_{t+j}) \qquad (1)$$

R は片側の文脈窓の長さを表す。$V(w_t) \in \mathbb{R}^d$は単語 w_tを表す d 次元の単語ベクトルであり、中間層 $H_{CBOW} \in \mathbb{R}^d$は次元 d のベクトルである。

学習はネガティブサンプリング (Negative Sampling, NS) [5] によって行う。NS では、目的関数 J:

$$J = \sum_{w_t \in T_{all}} \Big(\log\big(\sigma\left(H_t W(w_t)\right)\big) + \sum_{w_i \in N_t} \log\big(\sigma(-H_t W(w_i))\big) \Big) \qquad (2)$$

を最大化するように学習が行われる。ただし t は学習開始から何回目のサンプリングなのかを示す変数で、T_{all}は学習データ中の全ての単語トークンを表す。中心単語が w_t のとき NS によって選ばれた k 個の負例の集合を N_t とする。ここで、正例は文脈の中心の単語 w_tを、負例は学習データ中に出現する w_t以外の全ての単語を指す。通常 w_t 以外の全単語を負例とみなすところを、k 回のランダムサンプリングによって負例を限定して選択する手法が NS であり、正例と負例をロジスティック回帰によって分類する。



(a) CBOW (b) LR (c) WO

図 2. 既存モデルと提案モデルの概要図

モデルは確率的勾配降下法 (Stochastic Gradient Descent, SGD) を用いて最適化し、ベクトルと重みベクトルの更新を 1 単語ずつ行う。つまり、式 (2) の t が 1 進むごとに W, V の更新を行う。また、ある単語 v が $t = u$ の文脈において尤もらしいのかを表すスコアを式 (3) のように定義する。

$$score(u, v) = H_u W(v) \tag{3}$$

このスコアが高ければ文脈の中心単語としてふさわしい度合いが高いと判断する。式 (3) では $v = w_u$が正例である。また、スコアの定義から、式 (2) は正例のスコアを大きくし、負例のスコアの和を小さくするように学習することを表していると言える。

3 提案モデル

CBOW は中心単語を予測するモデルでありながら語順を無視しているため、中心単語を予測して同義表現を付与するには精度的に問題がある。そこで、語順情報をベクトルに含む以下の 2 つのモデルを提案する。

単語の左右を区別した図 2(b) のモデルを Left and Right (LR)、文脈窓の語順を全て区別した図 2(c) のモデルを Word Order (WO) と呼ぶことにする。また、本稿では中心単語の前後の窓中の単語を文脈と定義する。

LR では文脈中で w_t より前の単語と後の単語で V を別々に足し合わせて得たベクトルを結合して中間層を得る。

$$\begin{aligned} \boldsymbol{H_{LRt}} &= (H_{left}\ ;\ H_{right}) \\ &= \left(\sum_{j=1}^{R} V(w_{t-j})\ ;\ \sum_{j=1}^{R} V(w_{t+j})\right) \end{aligned} \tag{4}$$

WO では文脈の単語ベクトルを全て順番に結合することで中間層を得る。

$$\begin{aligned} \boldsymbol{H_{WOt}} &= (H_{t-R}; \cdots ; H_{t-1}; H_{t+1}; \cdots ; H_{t+R}) \\ &= (V_{t-R}; \cdots ; V_{t-1}; V_{t+1}; \cdots ; V_{t+R}) \end{aligned} \tag{5}$$

$H_{LR} \in \mathbb{R}^{2d}$は次元 $2d$、$H_{WO} \in \mathbb{R}^{2dR}$は次元 $2dR$ のベクトルである。例として図 2 では、$d = 3,\ R = 2$ なので、それぞれ H_{CBOW}は 3 次元、H_{LR}は 6 次元、H_{WO}は 12 次元になる。重みベクトル W も H と同様に同じ大きさの次元で作成する。

提案モデルでのスコアは式 (6), (7) のように求められる。

$$\begin{aligned} \text{LR}: score(u, v) &= score_{left}(u, v) + score_{right}(u, v) \\ &= H_{left} W(v) + H_{right} W(v) \end{aligned} \tag{6}$$

$$\begin{aligned} \text{WO}: score(u, v) &= \sum_{-R \le j \le R, j \neq 0} score_{u+j}(u, v) \\ &= \sum_{-R \le j \le R, j \neq 0} H_{u+j} W(v) \end{aligned} \tag{7}$$

LR では文脈左右、WO では文脈の全ての位置についてのスコアを求めることが可能になり、単語予測の精度が改善されると考えられる。

4 実験

4.1 使用データ

本稿では、学習データ、開発データとして共に英語 Wikipedia の平文を利用し、そのうちの 20,000,000 文を学習データとして用いた。本データは 2013 年 11 月時点にダウンロードしたものである。データは単語分割を行ったのみで、小文字化等は行っていない。

4.2 実験設定

学習データ中での単語の出現頻度数が閾値を下回る単語を未知語とし、未知語には V, W ともゼロベクトルを与え、学習を行わなかった。本稿では単語の出現頻度の閾値を 10 として得た 408,488 種類の単語を語彙 D と定義し、全語彙に対して V, W を与えた。学習中の窓長 R は 5 で一定にし、窓が前後の行と被ってしまう部分は無視してゼロベクトルで補った。

本稿では、窓をかける際に中心単語 w_t の文章中での出現頻度 $f(w_t)$ に応じた以下の式で得られる $p(w_t)$ の確率で学習を行わない手法 (サブサンプリング [5]) を取り入れた。

$$p(w_t) = 1 - \sqrt{t / f(w_t)} \tag{8}$$

このサブサンプリングは “the” や “to” 等の出現頻度が高い単語が過剰に学習されるのを防ぐと同時に、計算時間の短縮に貢献している。

他のパラメータは次元 $d = 50$, NS の回数 $k = 5$ とした。

4.3 学習率のスケジューリング

word2vec では学習率を初期値からゼロに向けて線形に下げる手法を用いているが、この手法だと学習率の下がり方が学習データの大きさに依存してしまうため、本稿では Bottou が推奨している、以下の式 (9) に従って学習率を下げる手法 [2] を使用した。

$$\varepsilon_t = \frac{\varepsilon_0}{1 + \varepsilon_0 \lambda t} \tag{9}$$

ε_0 と λ は定数で、本稿では $\varepsilon_0 = 0.025, \lambda = 1.0 \times 10^{-7}$ とした。

4.4 定量的評価

提案モデルによる単語の予測精度を計測する目的で、ソフトマックス関数を用いて、文脈から正例 w_t が予測される対数尤度 l_t を求めた。

$$l_t = \log p_t = \log \left(\frac{\exp\left(score(w_t)\right)}{\sum_{w_i \in D} \exp\left(score(w_i)\right)} \right) \tag{10}$$

評価データ内で文頭から順に窓をかけて対数尤度 l_t を測定し、$t = 500$ 回の計測の平均を平均対数尤度 L とする。学習中に一定の間隔で L を学習データ、開発データ両者について測定した。ただし、文脈が 2 単語以下の場合と窓内に未知語を 2 つ以上含む場合は、l を求めず評価をスキップした。

4.5 単語の推測

学習済のベクトルを用いて、中心単語の意味を推測する。ある文脈における中心単語について、式 (3) のスコアを全語彙 D について求め、スコアが高い単語の上位を表示する。D の大きさが前述のとおり 408,488 と大きいので、本稿では予め同義語集合を既存のシソーラス [1] から選択しておき、文脈によって、中心単語は同義語集合中のどの意味に近いのかを表示する実験も行った。

5 結果と考察

5.1 尤度の比較

横軸に学習データ全体を何周学習したかを表す繰り返し回数、縦軸に平均対数尤度 L をとったグラフが図 3 と図 4 である。学習データにおいて L を測ったグラフが図 3、開発データにおいて L を測ったグラフが図 4 である。学習が進むに連れて、両データにおいて平均対数尤度の上昇がみられた。学習データにおいても L が上昇していることから、NSの目的関数 (2) は負例をわずか $k = 5$ 個に限定していながら、全語彙 D についてのスコアの和を分母にした式 (10) の尤度を上昇させていることが確認できる。また、CBOW モデルよりも LR モデル、WO モデルが高い尤度を得られており、提案モデルによる正例予測確率が既存モデルを上回っている。

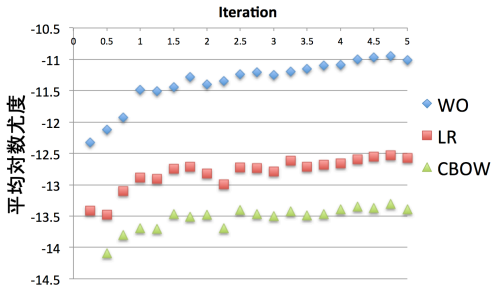


図 3. 学習データにおけるの尤度の比較

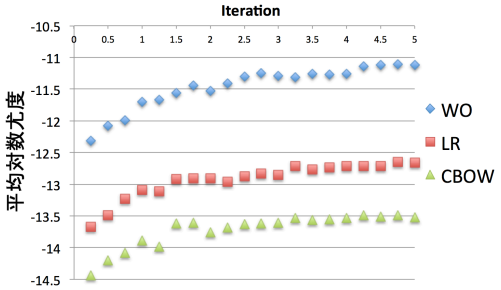


図 4. 開発データにおける尤度の比較

図 3 と図 4 を比較すると、開発データにおける尤度は学習データにおける尤度に比べ僅かに低いが、両者に大きな差異は無いことが分かる。これはつまり、学習に用いていないデータにおいても予測精度がある程度保たれていることを示している。WO による学習は、CBOW の学習と比較して 5 倍程の計算時間が必要になるが、計算を並列化することで本問題は解決されると考えられる。

5.2 単語の予測結果の例

表 1 の例の文脈は “The ski runs use the east face of the physical hill ,” で、w_t =“face” である。この w_t について、全語彙 D からスコアが最も高い単語 v_{1st} を探し出し、文脈中の単語の位置別にスコアを示している。CBOW の行の数字は式 (3) で求まるスコア、LR の行は式 (6) の $score_{left}(t, v_{1st})$ と $score_{right}(t, v_{1st})$、WO の行は $-5 \leq j \leq 5, j \neq 0$ における式 (7) の $score_{t+j}(t, v_{1st})$ を表示している。CBOW では語順情報が無視されるので、文脈の単語全体と似た単語が出力される。この例では v_{1st} =“ski” であったが、中心単語 “face” の位置に来る単語としては意味的に不適切である。LR は v_{1st} =“slopes” を出力した。本モデルを利

[1] http://thesaurus.com

表 1. ある文脈において各モデルが予測したスコアが最も高い単語とスコアの内訳

モデル	v_{1st}	$t-5$	$t-4$	$t-3$	$t-2$	$t-1$	t	$t+1$	$t+2$	$t+3$	$t+4$	$t+5$
		ski	runs	use	the	east	**face**	of	the	physical	hill	,
CBOW	**ski**	9.145										
LR	**slopes**	6.368					×	1.168				
WO	**site**	-0.029	-0.265	0.525	0.491	1.062	×	0.527	0.748	0.791	2.368	1.115

表 2. 【 Karl Marx became a leading **figure** in the International and a 】 member of its General Council . is not expected to increase .

同義語	スコア	上位単語	スコア
symbol	2.009	researcher	9.446
total	0.586	conductor	8.519
cipher	0.313	economist	8.251
character	0.130	investor	8.191

表 3. 【 This **figure** is not expected to increase 】 significantly so long as the land border between the Armenia and Turkey remains closed .

同義語	スコア	上位単語	スコア
integer	0.609	disadvantage	7.030
digit	0.201	discrepancy	6.303
number	0.025	generalization	6.022
cost	-0.111	capability	5.913

用することで、$score_{left}(t, v_{1st})$ が大きいので “slopes” の左側に “ski, runs, use, the, east” という単語が出現する確率が高い、といった考察が可能になる。WO では v_{1st} =“site” であった。この例だと、$score_{t-1}, score_{t+4}$ が高いことから、“site” の 1 文字前に “east”, 4 文字後に “hill” が位置している確率が高いことが言える。WO は語順を考慮しているので、文脈中の各単語との共起情報が分かるようになり、既存モデルよりも正確で具体的な同義表現の付与が可能になる。LR, WO モデルによって得られた v_{1st} は例中の “face” と入れ替えても意味的に違和感がなく、提案モデルによって単語の予測精度が向上していることが言える。

表 2 と表 3 は、WO で学習したベクトルを用いて、それぞれの例文中における単語 w_t =“figure” について、同義語集合中のスコア上位 4 単語を左側に、全語彙 D 中の同じくスコア上位 4 単語を右側に示している。なお、例文中の括弧は文脈窓を表している。同義語集合は第 4.5 節で述べたように既存のシソーラスを用いて作成した。表 2 では同義語セット中で “symbol” のスコアが、と表 3 では “integer” のスコアが最も高くなっており、文脈に見合った適切な単語が予測されていると考えられる。表 2 の v_{1st} は “researcher” で、この例では適切な同義表現のスコアが高くなっているが、表 3 では窓が前の行と被ってしまい文脈の単語数が少なくなるためか、やや不適な単語が上位に来ているので、シソーラスを用いた同義語集合の作成が有効であると言える。

6 おわりに

本稿では、word2vec の CBOW モデルを改良した新たなモデルを提案し、提案モデルによって単語の予測精度が向上することを示し、多義語の曖昧性解消にも応用可能であることを明らかにした。今後は並列化による高速化や、実際のデータセットでの評価を行った上で、単語の予測精度を高めたい。

参考文献

[1] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pp. 137–186. Springer, 2006.

[2] L. Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*, pp. 421–436. Springer, 2012.

[3] S. Clinchant and F. Perronnin. Aggregating Continuous Word Embeddings for Information Retrieval. In *CVSC*, pp. 100–109. ACL, 2013.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *ICLR*, 2013.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger eds., *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013.

[6] Q. Zhang, J. Kang, J. Qian, and X. Huang. Continuous Word Embeddings for Detecting Local Text Reuses at the Semantic Level. In *the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pp. 797–806. ACM, 2014.