Semantic Autoencoderを用いた低頻度語埋め込みの生成

城内 聡志　　菊井 玄一郎
岡山県立大学

{cd30021r, kikui}@cse.oka-pu.ac.jp

1 はじめに

低次元のベクトルによって単語の意味を表す分散単語埋め込みは，多くの自然言語処理で使われている．しかし，ベクトルを作成するには大規模なコーパスを必要としており，コーパスが小さいときや，低頻度語を埋め込む際には性能が低下することが知られている．これは数回単語の使用例を見ただけで新しい単語の概念を習得できる人間の能力とは対照的である．この問題に対して，低頻度語の埋め込みに向けたskip-gramの拡張 [7]，文脈語の単語ベクトルの平均を単語ベクトルとする手法等が提案されている．その中でも，対象語の文脈ベクトルに線形変換を適用して単語ベクトルを取得するA La Carte Embedding[3]という手法は良い単語ベクトルが構築できる．しかし，これを実現するためには，線形変換の学習データとして十分な量の単語ベクトルとその文脈語のベクトルが必要となる．そこで，特徴量から推定量の線形変換を学習する際の過学習を防ぐ方法としてSemantic Autoencoder[4]という方法がある．

Semantic Autoencoderは線形回帰モデルと違い，意味的表現を予測するだけでなく，射影した意味空間から特徴空間を再構築するという制約がある．これにより汎化能力が高くなることがわかっている [4]．

A La Carete Embeddingの線形変換を行っている部分を，本研究では文脈ベクトルに行列を加算することで単語ベクトルへと変換するSemantic Autoencoderを用いて少量のデータの場合でも，低頻度の良い単語埋め込みを求める．

2 問題の定式化

語彙 V 内の単語 w の文脈で構成されているテキストコーパスを C_V とし，文脈は V 内の単語列で構成されている．広く使われているアルゴリズムで学習した単語埋め込みは $v_w \in R^d$ とする．

この研究の目標は，未知語 f の文脈特徴の集合 C_f を用いて埋め込み $v_f \in R^d$ を構成することである．f は未知語なため，C_f の数は $C_w(w \in C)$ の数に比べ極めて少ない．

3 既存手法

この節では，提案手法のベースとなる文脈語のベクトルを加算する加算法を示し，次にこれを線形変換するA La Carte Embeddingについて述べる．

3.1 加算法

未知語のベクトルを取得する方法として，未知語の文脈に出現する語の単語ベクトルの平均を未知語のベクトルとするアプローチを加算法と呼び，$v_f^{addtive}$ と表す．

$$v_f^{addtive} = \frac{1}{|C_f|} \sum_{c \in C_f} \frac{1}{|c|} \sum_{w \in c} v_w \qquad (1)$$

文脈語のベクトルを加算する手法は文を表現する際によく使用されている．また，未知語のベクトルとしても使えること確認されている [6]．しかし加算法はストップワード由来の成分を含んでおり，それらをベクトルから除去する必要がある．既存研究ではその問題に対し，ストップワードの除去 [6]，[7] やベクトルの主成分を削除する手法 [10] 等を用いている．しかしこの方法は上手くノイズを消去できなかったり，必要な情報を削除してしまったりする可能性がある．

3.2 A La Carte Embedding

A La Carte Embeddingではノイズの除去を線形変換として行う．具体的にはまず (2) 式のように，学習済みの単語ベクトル v_w を加算法による文脈埋め込み

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.

$v_w^{addtive}$ から復元できるような行列 $A \in R^{d\times d}$ を求める．[1]

$$v_w \approx Av_w^{Addtive} = A(\frac{1}{|C_f|}\sum_{c\in C_w}\sum_{w'\in c} v_{w'}) \quad (2)$$

学習した行列 $A \in R^{d\times d}$ を用いて文脈から単語埋め込みと同じ意味空間への変換は次の式で表せる．

$$v_f = Av_f^{Addtive} = A(\frac{1}{|C_f|}\sum_{c\in C_f}\sum_{w\in c} v_w) \quad (3)$$

なお，行列 $A \in R^{d\times d}$ を求めるための線形回帰は次の式になる．

$$\underset{A\in R^{d\times d}}{\operatorname{argmin}} \sum_{w\in V} \|v_w - Au_w\|_2^2 \quad (4)$$

線形回帰を行う際の過学習を抑止するために，l_2 正則化や，単語 w の出現回数に応じた重みづけが用いられている．

4 提案手法

2 つの制約により A を推定する．1 つは特徴空間から意味空間へ射影し，その意味空間から特徴空間を再構築する Semantic Autoencoder を用いること (図 1)．もう 1 つは対象語のベクトルと文脈特徴のベクトルとの関係の空間に文脈特徴を射影すること．(5) 式で 2 つの制約から行列 A を求める．

$$\underset{A\in R^{d\times d}}{\operatorname{argmin}} \sum_{w\in V} \|u_w - A^*Au_w\|_2^2 \quad s.t. Au_w = v_w - u_w \quad (5)$$

ここで，$A^* = A^T$ という制約をつけることで，汎化能力向上のための l_2 正則化項を付ける必要がなくなる [2]，[4]，[1]．

Semantic Autoencoder は線形回帰の目的関数と制約条件を加重和した (6) 式に変換できる．

$$\min_{A\in R^{d\times d}} \sum_{w\in V} (\|Au_w - (v_w - u_w)\|_2^2 + \lambda\|A^T(v_w - u_w) - u_w\|_2^2) \quad (6)$$

λ は重みつけ定数である．

A La Carte Embedding における文脈ベクトルに行列 A をかけたものと単語ベクトルが同等であることは [5] で示されている．本研究で用いている Semantic Autoencoder では (6) 式の 1 つ目の項を文脈ベクトルと文脈ベクトルに行列 A をかけたものを足すことで単語ベクトルを構築する (8) 式に変形できる．

$$v_w = (I + A)u_w \quad (7)$$

(8) 式は，文脈ベクトルに行列 (I+A) をかけて単語ベクトルを構築しており，[5] の理論を提案手法にも適応できる．

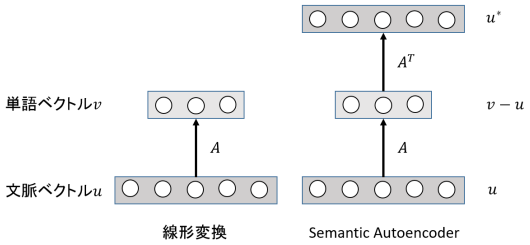

図 1: 線形変換と SAE の概要

5 実験

5.1 実験方法

評価は Contextual Rare Words(CRW)[3]，Nonce[7]，Chimera[6] という 3 つのタスクで行う．学習データの量による影響も確かめるために Nonce タスクで追加の実験を行う．

Contextual Rare Words: Rare Word(RW) データ [9] には希少語とそれに関連している単語のペアがいくつか用意されている．各ペアごとに 2 つの語の類似度を求め，類似度順にペアをソートする．類似度は，希少語については使用例から推定したベクトル，ペアのもう一方の語は Westbury Wikipedia Corpus(WWC) データ [8] の希少語の入った文を抜いたものから学習したベクトルを作り，これらの cos 値とする．このようにしてソートされたものと，人間がペアに類似度をつけてソートしたものとの spearman 順位相関係数を測る．評価は 1，2，4，...，128 の文脈例ごとに行う．

Nonce: Wikipedia で学習した単語ベクトル (gold ベクトル) を 1 つの使用例から再現できるかを測るタスクである．まず，Wikipedia のタイトルを抽出する．次に各タイトルのページの最初の文を抽出する．その際，文中のタイトルの単語列を nonce に置き換える．抽出したデータの中で ukWaC コーパス [11] にて 200 回以上出現しているタイトルを含む文を保持する．そこから 1000 文サンプリングし，700 文を検証用データに，

[1] $\frac{1}{|c|}$ は文脈のウィンドウサイズを固定すると省略できる．

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.

300 文をテストデータに分ける．そして，1 つの使用例のみから作られた nonce の単語ベクトルがどれだけ gold ベクトルに類似しているかを Wikipedia で学習した語彙 259,376 個中の類似度順位の Mean Reciprocal Rank(MRR) と Median Rank(MR) を用いて測る．線形変換をおこなう行列 A の学習に必要な文脈語には，テストデータの nonce の入った文以外の Wikipedia で Word2Vec を用いて学習した単語埋め込みと，その文脈単語を使用して構築する．

本研究では，文脈ベクトルから単語ベクトルへ変換する関数を学習するデータが少ない場合，A La Carte Embedding に比べて Semantic Autoencoder が良い性能であるかを確認するために，データを減らした時の性能の減少具合も確認する．

2 つの関連した単語（例：gorilla と bear) を結合したものを chimera とする。chimera の文脈情報としてもとの単語が出現する文脈を同じ数だけ集めてこれらの単語を同じ nonce に置き換えたものを用意する。この文脈情報から chimera 語の意味を推定し、別に与えられた 6 語程度をこの chimera 語との類似性の順に並べる。人間による並びと推定された chimera 語のベクトルとのコサイン類似度による並びとの spearman 相関で評価する。

Chimera: 2 つの関連した単語（例:gorilla と bear) を結合したものを chimera とする。chimera の文脈情報としてもとの単語が出現する文脈を同じ数だけ集めてこれらの単語を同じ nonce に置き換えたものを用意する。この文脈情報から chimera 語の意味を推定し、別に与えられた 6 語程度をこの chimera 語との類似性の順に並べる。人間による並びと推定された chimera 語のベクトルとのコサイン類似度による並びとの spearman 相関で評価する。chimera は 33 個用意し，その使用例の文には British National コーパスと ukWaC コーパスからランダムに選び，結合前の単語を含む文を各 chimera に 2，4，6 文ずつ用意する．chimera との類似度を調べる単語はそれぞれ 6 つ選ぶ．選ばれた単語は Wikipedia で Word2Vec を用いて学習する．

3 つの手法を比較した.「加算法」,「A La Carte」は 2 節で説明したベースモデル.「A La Carte(SAE)」は 3 節で説明した提案手法であり，Semantic Autoencoder を用いて文脈から単語ベクトルを導出するモデルである．

5.2 実装詳細

「A La Carte」では線形回帰の部分を sklearn と theano の二種類の実装を行った.「A La Carte(SAE)」は theano で実装を行った．theano で実装したもののパラメータの最適化には SGD を用い，epoch 回数は 200 回，batch size は 10 とした．学習率は [0.05, 0.01, 0.005, 0.001] の中から，λ は [0.5, 1, 1.5] から検証用データのスコアが最も良くなるなるものを選んだ．Nonce タスクと Chimera タスクの単語ベクトルの次元は 400 とし，CRW タスクは 300 とした．Wikipedia で Word2Vec を用いて学習した単語ベクトルは [7] を用いた．

5.3 実験結果と考察

Nonce，Chimera タスクの結果を表 1 に示す．Chimera タスクでは既存手法を超えることはできていないが，Nonce タスクでは MRR，MR ともに向上しており，CRW タスクでも図 2 より，spearman 順位相関係数が向上していることがわかる．図 3 を見ると Nonce タスクにおいて，提案手法は既存手法に比べて少量の学習データでも文脈ベクトルから単語ベクトルが再現できている．これより，汎化能力の高い Semantic Autoencode によって少量のデータからでも上手く学習していることがわかる．

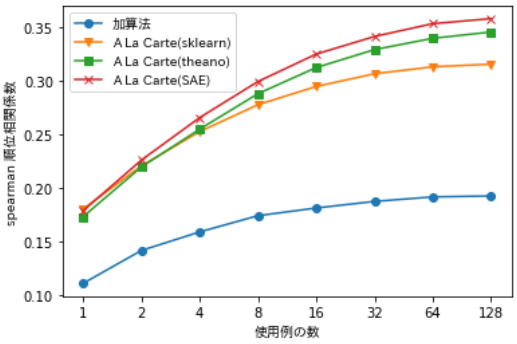

図 2: 使用例数の影響も考慮した Contextual Rare Words タスクにおける spearman 順位相関係数

6 おわりに

本研究では，Semantic Autoencoder を用いた未知語ベクトルの推定手法を提案した．この手法は，学習データが少ない場合でも，十分な量がある場合と比較して同等の性能をもつ単語ベクトルを構築できること

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.

表 1: 提案手法と既存手法の比較

手法	Nonce		Chimera		
	Mean Reciprocal Rank	Median Rank	2 文	4 文	6 文
加算法	0.00945	3381	0.3627	0.3701	0.3595
a la carte(sklearn)	0.07058	165.5	0.3634	0.3844	0.3941
a la carte(theano)	0.07716	114	0.3591	0.3911	0.4004
a la carte(Semantic Autoencoder)	0.07803	96	0.3738	0.3729	0.3964

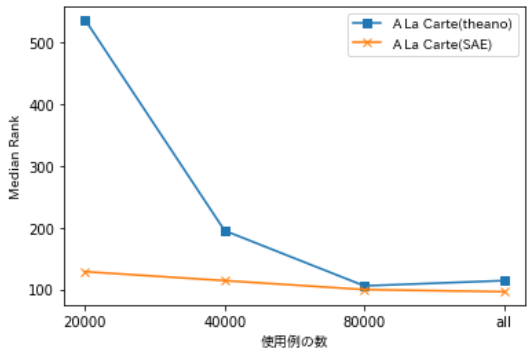

図 3: Nonce タスクの学習データの数による影響

がわかった．学習済みのベクトルとどれだけ近いかを測る Nonce タスクでは，MRR，MR が向上しており，CRW タスクでも，spearman 順位相関係数が向上した．今後の課題として，希少語の構成文字列の情報を使用することがあげられる．

参考文献

[1] Hong-You Chen, Cheng-Syunan Lee, Keng-Te Liao. Word Relation Autoencoder for Unseen Hypernym Extraction Using Word Embeddings. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4834-4839, 2018.

[2] Marc'Aurelio Ranzato, Y-Lan Boureau, Yann LeCum. Sparse Feature Learning for Deep Belief Networks. In Advances in Neural Informetion Processing Systems 20, pages 1185-1192, 2008.

[3] Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, Snjeev Arora. A La Carte Embedding: Cheap but Effective Induction of Semantic Feature Vectors. In Proceedings of the 56th Annual Meating of the Association for Computational Linguistics, pages 12-22, 2018.

[4] Elyor Kodirov, Tao Xiang, Shaogang Gong. Semantic Autoencoder for Zero-Shot Learning. In Computer Vision and Pattern Recognition, Computer Vision Foundation, 2017.

[5] Sanjeev Arora, Yuanzhi, Yingyu Liang, Tengyu Ma, Andrej Riteski. Linear Algebraic Structure of Word Sences, with Applications to Polysemy. In Transactions of the Association for Computational Linguistics, vol.6, pages 483-495, 2018.

[6] Angeliki Lazaridou, Marco Marelli, Marco Baroni. Multimodal Word Meaning Induction From MinimalExposure to Natural Text In Cognitive Science 41, pages 677-705, 2017.

[7] Aurelie Herbelot, Marco Baroni. High-risk learning: acquiring new word vectors from tiny data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 304-309, 2017.

[8] Cyrus Shaoul and Chris Westbury. The westbuly lab wikipedia corpus.

[9] Minh-Thang Luong, Richard Socher, Christopher D.Manning. Better Word Representations with Recursive Neural Networks for Morphology. In Procerdings of Seventeenth Conference on Computational Natural Language Learning, pages 104-113, 2013.

[10] Sanjeev Arora, Yingyu Liang, Tengyu Ma. A Simple but togh-to-beat baseline for sentence embeddings. In Proceedings of the 5th International Conference on Learning Representation, 2017.

[11] Marco Baroni, Silvia Bernardini, Adriano Ferraresi, Eros Zanchetta. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. Language resources and evaluation, pages 209-226, 2009.

Copyright(C) 2019 The Association for Natural Language Processing.
All Rights Reserved.