Vol.2018-ICS-192 No.9
2018/7/6

反義関係を反映する単語ベクトル変換手法の検討

別所克人[†1]　浅野久子[†1]　富田準二[†1]

概要：word2vec を始めとして、単語の意味概念を表し、単語間の意味的類似性を定量的に計ることができる単語ベクトルがこれまでに提案されている。これらの手法の多くは、単語の周辺分布に関する仮説をベースとしているため、反対の意味をもつ反義語のベクトルが近くなるという課題がある。本稿では、生成済みの単語ベクトルのセットである概念ベースが与えられたとき、反義語辞書中の反義語のベクトルはより遠くなるように、概念ベース中の全単語ベクトルを変換する手法を提案する。提案手法は、反義語辞書中にない反義語のベクトルも遠くなるという効果ももつ。4 つのタスクにおいて従来手法との比較実験を行った結果、2 タスクにおいては有意差が認められなかったが、残る 2 タスクにおいて提案手法は有意に高精度となり、提案手法による単語ベクトルの配置が、単語間の類似性をより反映したものとなることを報告する。

A Study of Word Vector Conversion Method Reflecting Relationships of Antonyms

KATSUJI BESSHO[†1]　HISAKO ASANO[†1]
JUNJI TOMITA[†1]

1. はじめに

単語の意味概念を表し、単語間の意味的類似性を定量的に計ることができる単語ベクトルとして、PLSA[1]や word2vec[2], GloVe[3], fastText[4]等が提案されている。これらの手法により、各単語が n 次元ベクトルで表現され、意味的に近い単語のベクトルは近くに配置され、単語間の意味的類似性を、対応する単語ベクトル間の距離で算出することができる。これらの手法は、「意味的に似ている単語は、コーパス中のその周辺文脈における単語の頻度分布も似ている傾向がある」という分布仮説[5]をベースとして、各単語に対し、コーパス中のその周辺文脈をもとに単語ベクトルを生成している。本稿では、単語とそのベクトルとの対のセットを概念ベースと呼ぶことにする。

生成した概念ベースを用いて、テキスト間の類似性を表す距離を算出することができる。例えば任意のテキストに対し、テキスト中の単語のベクトルの重心を、該テキストのベクトルとする。テキスト間の距離を、対応するテキストベクトル間の距離として算出する。これは単語ベクトルの最もシンプルな適用例だが、テキスト検索やテキスト分類、DNN を用いた学習・推定等、言語処理の広範囲において、単語ベクトルを用いることが現在、普通になっている。

「高い、安い」といった反対の意味をもつ反義語に関し、そのベクトル間の距離が近いことは、ベクトルから意味概念を識別することが困難になるため好ましくない。しかし、反義語の周辺文脈は似かよっているため、分布仮説をベースとする手法で生成した反義語のベクトルは近くなるという課題がある。

このことによりテキストベクトル間の距離関係も不適切なものとなる。例えば、反義語「高い、安い」のベクトル間の距離が近すぎる場合、「高い」に対し、反義語「安い」の方が、同義語「高価」よりもベクトル間の距離が小さくなる。このため、以下のテキスト X に対し、テキスト Z の方がテキスト Y よりも意味が近いにも関わらず、テキスト Y の方がテキスト Z よりもベクトル間の距離が小さくなる。

テキスト X：高いワイン
テキスト Y：安いワイン
テキスト Z：高価なワイン

このように、単語ベクトルを活用する様々な言語処理において、反義語を含み意味的に遠いテキストが、ベクトル表現としては不当に近く識別性が低いものとなり、このことが精度低下の一因となっている。

本来、「高い」に対し、反義語「安い」の方が、同義語「高価」よりもベクトル間の距離が大きくあるべきである。そうなっていれば、テキスト X に対し、テキスト Y の方がテキスト Z よりもベクトル間の距離が大きくなる。このように、反義語のベクトル間の距離を大きくし、反義語のベクトルの識別性を高める必要がある。

本稿では、上記課題を解決するために、生成済みの概念ベースが与えられたときに、反義語のペアを格納した反義語辞書を参照し、反義語辞書中の反義語のベクトルはより遠くなるように、概念ベース中の全単語ベクトルを変換する手法を提案する。提案手法は、反義語辞書中にない反義語のベクトルも遠くなるという効果ももつ。

以下、2 節で関連研究について述べ、3 節で提案手法を述べる。4 節で 4 つのタスクに関する評価実験について述べ、5 節でまとめを述べる。

†1 日本電信電話株式会社　NTT メディアインテリジェンス研究所
NTT Media Intelligence Laboratories,
Nippon Telegraph and Telephone Corporation

2. 関連研究

反義関係を始めとする単語間関係に関する外部知識を、単語ベクトルに反映させる手法として、単語ベクトルの生成過程で外部知識を反映するように単語ベクトルを生成する手法（[6],[7]）と、一旦、単語ベクトルを生成した後に外部知識を反映するように単語ベクトルを変換する手法（[8],[9],[10]）がある。後者の手法は、多様な単語ベクトル生成手法に依存せず適用できるという利点がある。提案手法は後者の手法である。

[6]では、類義語のペアに対しては類似度が大きい程、反義語のペアに対しては類似度が小さい程、値が大きくなる項を Skip-Gram with Negative Sampling と組み合わせた目的関数を最大化する単語ベクトルを求める手法が提案されている。

[7]では、類義語間の類似度は反義語間の類似度より大きいといった、外部知識から得られるペナルティ項を skip-gram model に組入れた目的関数を最大化する単語ベクトルを求める手法が提案されている。

[8]では、以下の目的関数$\psi(Q)$を最小化する変換後単語ベクトルを求める retrofitting という手法が提案されている。この最適化は、変換後単語ベクトルq_iと変換前単語ベクトル$\hat{q}_i$とが近くなるように、q_iと同義語・類義語等の変換後ベクトルq_jとが近くなるようにするものである。

$$\psi(Q)=\sum_{i=1}^{n}\left[\alpha_i\left\|q_i-\hat{q}_i\right\|^2+\sum_{(i,j)\in E}\beta_{ij}\left\|q_i-q_j\right\|^2\right]$$

[9]では、ConceptNet を外部知識とし拡張された retrofitting を行い、SemEval-2017 Task2 の多言語単語類似度の評価で最優秀の結果を出している。

[10]では、後述する目的関数$C(V,V')$を最小化する変換後ベクトルを求める counter-fitting という手法が提案されている。変換前単語ベクトル群を$V=\{v_1,v_2,\cdots,v_N\}$、変換後単語ベクトル群を$V'=\{v'_1,v'_2,\cdots,v'_N\}$とし、$d(v_i,v_j)=1-\cos(v_i,v_j)$, $\tau(x)=\max(0,x)$とする。

$$AR(V')=\sum_{(u,w)\in A}\tau(\delta-d(v'_u,v'_w))$$

A:反義語ペアセット，$\delta=1.0$

$$SA(V')=\sum_{(u,w)\in S}\tau(d(v'_u,v'_w)-\gamma)$$

S:同義語ペアセット，$\gamma=0.0$

とおき、上記以外の単語ペアに関し、

$$VSP(V,V')=\sum_{i=1}^{N}\sum_{j\in N(i)}\tau(d(v'_i,v'_j)-d(v_i,v_j))$$

$N(i)$:単語iから半径ρ内の単語，$\rho=0.2$

とおく。以下の目的関数$C(V,V')$を最小化する変換後単語ベクトル群V'を求める。

$$C(V,V')=k_1AR(V')+k_2SA(V')+k_3VSP(V,V')$$

$$k_1=k_2=k_3\geq 0$$

上記で各パラメーターの値は、[10]の実験で用いた値である。[10]では、英語共通データセットである SimLex-999 データセットを用いた単語ペア群の類似スコアによるランキングタスクにおいて、retrofitting の手法[8]よりも高精度であったことを報告している。

3. 提案手法

提案手法は、生成済みの概念ベースが与えられたときに、反義語のペアを格納した反義語辞書を参照し、反義語辞書中の反義語のベクトルはより遠くなるように単語ベクトルを変換するものである。ただ、それだけを行うと、反義語辞書中にない任意の単語ペアについては、ベクトル間の距離が不当に大きくなったり小さくなったりし、配置が適切なものでなくなる。このため提案手法は、反義語辞書中にある反義語のベクトル間距離をより遠くするのと同時に、反義語辞書中にない単語ペアのベクトル間距離は可能な限り変化がないように、概念ベース中の全単語のベクトルを変換する。

すなわち提案手法では、概念ベース中の任意の単語 A, B のペア C に対し、C が反義語辞書にある場合、A, B の変換後ベクトル間の距離d'と、A, B の変換前ベクトル間の距離dに値$\alpha(>0)$を加算した値とが可能な限り等しくなり、かつ、C が反義語辞書中にない場合、d'とdとが可能な限り等しくなるように、概念ベース中の全単語のベクトルを変換する。これは以下のように定式化される。

概念ベース中の単語のリストを、$W_1,W_2,\cdots,W_m$とする。単語W_iの変換後ベクトルをω'_i、変換前ベクトルをω_iとす

る。単語対W_i, W_jに対し、変換後ベクトル間の距離を$d'_{\{i,j\}} = \left\| \omega'_i - \omega'_j \right\|$とし、変換前ベクトル間の距離を$d_{\{i,j\}} = \left\| \omega_i - \omega_j \right\|$とする（距離は L2 ノルムである）。以下の目的関数Fを最小化する各単語の変換後ベクトルのリストである行列$\left(\omega'_{pq}\right)$を求める。

$$F = \sum_{\{i,j\} \in \{\{i,j\} | 1 \le i < j \le m\}} F_{\{i,j\}} = \sum_{\{i,j\} \in \{\{i,j\} | 1 \le i < j \le m\}} \left(d'_{\{i,j\}} - \left(d_{\{i,j\}} + \alpha_{\{i,j\}} \right) \right)^2$$

$\alpha_{\{i,j\}}$は、$\{i,j\}$に依存する値で、あらかじめ定めておく。W_i, W_jが反義語辞書中にある場合、$\alpha_{\{i,j\}} > 0$とし、反義語辞書中にない場合、$\alpha_{\{i,j\}} = 0$とする。

目的関数Fを最小化する行列$\left(\omega'_{pq}\right)$を、最適化手法の一つである AdaGrad[11]を用いて求める。具体的には一つのターンにおいて、各W_xに対し、W_xの反義語辞書中の反義語W_yに対する$F_{\{x,y\}}$に関する$\left(\omega'_{pq}\right)$の更新計算をした後、計算量低減のため、$W_x$との変換前ベクトル間距離が上位 N 位以内の単語$W_y$に対してのみ$F_{\{x,y\}}$に関する$\left(\omega'_{pq}\right)$の更新計算を行う。但し、一つのターンにおいて、集合$\{x,y\}$に関する更新計算は 1 回のみとする。このターンを所定の回数 L だけ行う。

このようにして、概念ベース中の各単語とその変換後ベクトルとの対のセットである変換後概念ベースが生成される。

提案手法は、単語ペアの属性に応じて、その変換後ベクトル間の距離を調整するという点で、counter-fitting の手法と同様である。だが、数式において以下の差異がある。

couter-fitting は、ベクトル間距離を、1 からコサイン類似度を減じたものとしている。また、反義語の変換後ベクトル間の距離は、δ 以上であれば、どれだけ大きくてもよく、同義語の変換後ベクトル間の距離は、γ 以下であれば、どれだけ小さくてもよく、それ以外の単語ペアの変換後ベクトル間の距離は、変換前ベクトル間の距離以下であれば、どれだけ小さくてもよい。

一方、提案手法は、ベクトル間距離を L2 ノルムとしている。また、単語ペアの変換後ベクトル間の距離は、変換前ベクトル間の距離に、ある値を加算した値の近傍内にあるようにする。これにより、反義語が極端に遠くなったり、反義語・同義語でない単語ペアが極端に近くなったりするのを抑制する。この結果、反義語辞書中の反義語ペアのベクトルは適度に離れた位置にあるようになり、また、それ以外の単語ペアのベクトル間距離はなるべく変換前の距離を維持するようになり、任意の単語ペアのベクトル間距離が適切となっている配置になることが期待できる。

4. 評価実験

4.1 概要

提案手法による変換後概念ベースについて以下の 4 つのタスクの評価実験を行った。

- 単語連想
- 関係単語検索
- 単語ペアランキング
- 言い換え文検索

最初の 3 つのタスクで、変換後単語ベクトルの配置の妥当性を評価する。また、それとは別に、変換後単語ベクトルを何らかの応用タスクに適用した場合の精度を評価する。今回はその応用タスクとして、4 つ目のタスクである言い換え文検索を採用した。

単語ペアランキング以外のタスク評価は日本語データで行い、単語ペアランキングは英語データで行った。

比較手法として、counter-fitting[10]と retrofitting[8]をとった（但し、単語ペアランキングでは、[10]において、counter-fitting の retrofitting に対する優位性が示されているので、counter-fitting のみを比較手法とした）。

以下、日本語データにおいて使用した概念ベース・反義語辞書と各手法のパラメータ値を述べる（英語データについては 4.4 節で述べる）。

概念ベース生成元のコーパスとして、Web 上の QA サイトから収集した 4,900,096 文書をとり、これを形態素解析器 JTAG[12]により形態素解析し、名詞・動詞・形容詞等の内容語のみとした（活用語は終止形とした）。この結果、単語延べ数は 911,446,805 となった。この形態素解析結果から word2vec ベクトルを生成した。word2vec ベクトル生成コマンドのオプションは、-size：100, -window：5, -iter：100, -min-count：5 を指定し、他のオプションはデフォルト値とした。生成した word2vec ベクトルは長さに著しい差があり（最小値：0.027, 最大値：80.200）、ベクトル間距離に基づく単語間類似性の精度の低下をもたらすため、各ベクトルを長さ 1 に正規化した。これにより、335,040 個の長さ 1 の 100 次元単語ベクトルからなる概念ベースを生成した。

構成単語が概念ベース中にあるような反義語ペアを 6,281 個格納した反義語辞書を作成した。反義語辞書中の

単語の異なりは 10,161 個であった。図 1 は、反義語辞書中の一部の単語の異なりごとに、その反義語をリストしたものである。

単語	反義語のリスト
高い	低い　安い
寒い	暖かい　暑い
降りる	乗る　登る　上がる
学生	先生　教師　教員　社会人

図 1：反義語辞書の一部

提案手法のパラメータ値として、W_i, W_j が反義語辞書中にある場合、任意の $\{i, j\}$ に対し $\alpha_{\{i,j\}} = 4.0$ とし、N=100, L=5 とした。

counter-fitting のプログラムは、[13]内のものを使用し、パラメータ値は、[13]内の設定ファイル中の値を使用した。

retrofitting のプログラムは、Adgrad の手法で実装した。目的関数 $\psi(Q)$ において、$\alpha_i = 1$ とし、$\{j \mid (i, j) \in E\}$ として W_i の反義語辞書中の反義語 W_j の集合をとり、この集合の要素数を γ_i としたとき、$\beta_{ij} = -1/\gamma_i$ とした。また、ターン回数を 2 とした。

変換前概念ベース, counter-fitting, retrofitting, 提案手法による変換後概念ベースを比較評価する。

以下、各タスクの評価実験について述べる。

4.2 単語連想の評価実験

基点となる単語（以下、基点語）に対し、その近傍（距離の近い M 個の単語）を導出することを本稿では単語連想と呼ぶ。基点語の近傍において、基点語の関連語の割合が大きい程、変換後単語ベクトルが、単語間の類似性を反映した好ましい配置をしているといえる。単語連想の評価では、基点語ごとに、その近傍における基点語にとっての関連語、反義語辞書に登録済みの反義語、反義語辞書に未登録の反義語、非関連語の割合を算出し、その平均を見ることとした。

基点語の選択は以下のようにした。反義語辞書中の単語で、概念ベース生成元コーパス中の出現頻度が 80 以上で、かつ、文字数が 2 以上のものを選択した。選択した各単語 A に対し、単語 A の変換前概念ベース中の近傍単語 30 個における単語 A の反義語辞書中の反義語の個数を算出し、算出した反義語の個数の多いものから 154 個の単語をとり、基点語とした。算出した反義語の個数は、4 基点語が 4 個、27 基点語が 3 個、123 基点語が 2 個であった。

概念ベースごとに、各基点語に対し、近傍単語 30 個を導出した。各近傍単語に対し、基点語にとっての反義語辞書に登録済みの反義語か、反義語辞書に未登録の反義語か、それ以外の何らかの関連がある関連語か、全く関連の無い非関連語かのラベルを付与した。表 1 は、各基点語の近傍における関連語、登録済反義語、未登録反義語、非関連語の割合の平均を示したものである。

	関連語	登録済反義語	未登録反義語	非関連語
変換前	42.3%	7.4%	17.8%	32.6%
counter	43.5%	0.0%	8.7%	47.7%
retro	46.3%	1.1%	14.1%	38.5%
提案手法	47.4%	0.3%	14.4%	37.9%

表 1：基点語の近傍における各単語種別の割合の平均

いずれの手法でも、登録済反義語の割合は変換後、0 近くになった。

一つの基点語に対し、登録済反義語と未登録反義語は意味が近く、変換前の距離が近い傾向にある。いずれの手法でも、登録済反義語が基点語から遠ざかるにつれ、未登録反義語も基点語から遠ざかり、未登録反義語の割合が変換後、小さくなった。

いずれの手法でも、非関連語の割合は変換後、大きくなるが、提案手法は従来手法より、非関連語の割合が小さい。特に counter-fitting では、非関連語を過剰に基点語に引き寄せてしまう傾向がある。

結果、提案手法は従来手法より関連語の割合が大きくなった。関連語の割合の平均に関する有意差検定の p 値は、提案手法と counter-fitting 間で 2.4×10^{-2} %, 提案手法と retrofitting 間で 3.0%であり、有意水準 5%で有意差が認められた。

図 2 は、変換前、及び、各手法での基点語「貸し家」に対する近傍単語とラベル付与結果を示したものである。○は関連語を、×は非関連語を示す。

変換前には上位にきていた反義語が、提案手法では順位を落としている。また、従来手法では非関連語の混入が目立つが、提案手法は非関連語の混入をなるべく抑制している。

4.3 関係単語検索の評価実験

word2vec のような単語分散表現では、アナロジータスクの評価により、同一の関係性にある単語ペアの各単語のベクトルの差ベクトルは、ほぼ同一のベクトルであるという性質があることが報告されている[2]。すなわち、単語 a のベクトルを U_a としたとき、同一の関係性にある単語ペア

	変換前			counter-fitting			retrofitting			提案手法	
1	ハイツ	○	1	区分所有管理士	×	1	ハイツ	○	1	ハイツ	○
2	マンション暮らし	×	2	mike	×	2	マンション暮らし	×	2	マンション暮らし	×
3	一軒家	○	3	ハイツ	○	3	一軒家	○	3	団地	○
4	賃貸し	○	4	xmonth	×	4	団地	○	4	一軒家	○
5	貸家	○	5	アオボウシインコ	×	5	貸家	○	5	コーポ	○
6	団地	○	6	ペット・サウンズ	×	6	一人住まい	○	6	一人住まい	○
7	住まう	○	7	machi	×	7	コーポ	○	7	一戸建	○
8	家持ち	反義（未）	8	服部緑地	○	8	ぼろ家	×	8	賃貸し	○
9	ぼろ家	×	9	duke	×	9	住まう	○	9	ぼろ家	×
10	一戸建	○	10	秀輝	×	10	家持ち	反義（未）	10	区分所有管理士	×
11	一戸建て	○	11	高石市	○	11	住人	○	11	貸家	○
12	一人住まい	○	12	マンション暮らし	×	12	区分所有管理士	×	12	住人	○
13	賃貸	○	13	伊丹市	○	13	一戸建	○	13	住まう	○
14	コーポ	○	14	せせらぎ	×	14	賃貸し	○	14	府営	○
15	アパート	○	15	park	×	15	一戸建て	○	15	戸建	○
16	マンション	○	16	bbking	×	16	府営	○	16	住む	○
17	住人	○	17	canoe	×	17	テラスハウス	○	17	近隣	○
18	借家	反義（登）	18	宜野湾	○	18	戸建	○	18	一戸建て	○
19	空き家	○	19	ゴー	×	19	ご近所	×	19	地内	○

図 2：基点語「貸し家」に対する近傍単語の様相

(a,b)と単語ペア(c,d)に対し、$U_b - U_a \approx U_d - U_c$が成り立つ。例えば、単語ペア$(\text{日本},\text{東京})$と単語ペア$(\text{中国},\text{北京})$は、国とその首都の関係性にあり、$U_{\text{東京}} - U_{\text{日本}} \approx U_{\text{北京}} - U_{\text{中国}}$が成り立つ。

各概念ベースが、この性質をどれだけ持っているかを評価した。同一の関係性にある単語ペア(a,b)と単語ペア(c,d)に対し、$U_b - U_a + U_c$から距離の近い順に概念ベース中の単語をランキングし、dの順位を導出する。これを本稿では関係単語検索と呼ぶ。dの順位が高い程、概念ベースがこの性質を持っていると評価できる。

評価用データとして、[14]の研究で作成された評価用データ[15]を利用した。評価用データ[15]は、関係性の種別ごとに、該当する英単語ペアのセットがある。本評価では、この内、表 2 で示した種別のデータを使用した。種別ごとに、単語ペアの全部ないし一部を和訳し、構成単語が概念ベース中にある単語ペアのみに限定した結果、表 2 の単語ペア数となった。各種別において、単語ペア(a,b)と単語ペア(c,d)に対し、タプル$U_b - U_a + U_c$に関する順位と、タプル$U_d - U_c + U_a$に関する順位を導出する。表 2 では、種別ごとの、順位導出対象タプルの数も示している。

概念ベースごとに、各種別の各タプルに関する順位を導出し、全種別の全タプルの導出順位の平均を算出した。結果を表 3 に示す。

いずれの手法も変換前より順位が落ちるが、提案手法は従来手法より順位が高く、単語ベクトル間の差分関係をなるべく崩さず維持している。有意差検定の p 値は、提案手法と従来手法間でほぼ 0%であり、有意差が認められた。単語連想の評価で、近傍単語における非関連語の割合は、従来手法の方が高かった。関係単語検索でも、従来手法においては、基点となるベクトルの近傍において、関係単語のベクトルよりも手前に非関連語のベクトルが、より多く配置されているものと考えられる。

4.4 単語ペアランキングの評価実験

[10]の実験で用いられた[13]内にある英語共通データセットの SimLex-999 データセットを用いて、単語ペア群の類似スコアによるランキングタスクの評価実験を行った。

概念ベースは、[13]内にある生成済みの GloVe 概念ベース（76,855 個の長さ 1 の 300 次元単語ベクトルからなる）を用いた。

反義語辞書は、[13]内にある PPDB 由来の反義語辞書と WordNet 由来の反義語辞書を用いた。構成単語が概念ベース中にある反義語ペアは重複分を除いて 6454 個となった。

提案手法のパラメータ値として、W_i, W_jが反義語辞書中にある場合、任意の$\{i,j\}$に対し$\alpha_{\{i,j\}} = 2.0$とし、N=100, L=11 とした。

counter-fitting のプログラムは、[13]内のものを使用し、パラメータ値は、[13]内の設定ファイル中の値を使用した。

関係性の種別	単語ペア(a, b)		単語ペア(c, d)		単語ペア数	タプル数
All capital cities	日本	東京	中国	北京	82	6642
Currency	日本	円	ロシア	ルーブル	112	12432
City-in-state	ロドニー	ミシガン州	ユリーカ	ユタ州	26	650
Man-Woman	父	母	紳士	淑女	36	1260
Nationality adjective	日本	日本人	フランス	フランス人	50	2450
Antonym	好き	嫌い	足し算	引き算	72	5112
MemberOf	ドイツ人	ドイツ	ブドウ	ブドウ科	14	182
MadeOf	涙	水	たばこ	ニコチン	44	1892
IsA	王妃	貴族	スケート	スポーツ	68	4556
SimilarTo	知覚	認識	羽	翼	14	182
PartOf	口	顔	サッシ	窓	97	9312
InstanceOf	ベルサイユ	宮殿	ガルブレイス	経済学者	42	1722
DerivedFrom	王	王国	目標	決意	29	812
HasContext	兵士	軍隊	バット	野球	41	1640
RelatedTo	試す	試験	争う	撃退	37	1332
Attribute	体重	軽い	音量	小さい	27	702
Causes	麻薬	幻覚	痛み	苦しむ	22	462
Entails	酷評	評価	習得	理解	44	1892
合計	/	/	/	/	857	53232

表 2：評価対象の関係性の種別

変換前	counter	retro	提案手法
30702.4	45935.9	39333.6	36097.9

表 3：手法ごとの順位の平均

SimLex-999 データセット中にある英単語ペアで、構成単語が概念ベース中にある 974 個の単語ペアを使用した。各単語ペアには、複数の作業者によって付与された類似スコアの平均が対応付けられている。単語ペア群を、各単語ペアのスコアの降順にソートし、正解ランキングとした。

概念ベースごとに、各単語ペアに、構成単語のベクトル間距離を対応付け、単語ペア群を、各単語ペアのスコアの降順にソートした。正解ランキングと出力ランキングとの間で、スピアマンの順位相関係数を算出した。結果を表 4 に示す。

変換前	counter	提案手法
0.42	0.53	0.52

表 4：手法ごとの順位相関係数

提案手法は、変換前よりは相関係数が高く有意差があった（p 値=0.6%）。しかし提案手法は counter-fitting より相関係数が低い結果となった。提案手法と counter-fitting 間で有意差検定を行った結果、p 値は 65.4%であり、この評価データに対しては、実質、有意差はないといえる。

SimLex-999 データセットには、以下の問題があると思われる。

SimLex-999 データセットの各単語ペアには 0 から 10 までの範囲の小数点以下 2 位までのスコアがつけられており、0 から 10 までの範囲を 1000 分割した値を付与している。これにより、本来意味的類似性の高さに序列を付けられない 2 つの単語ペアに序列が付く確率が高く、実際序列がついている例が散見され（例：「(alcohol, wine)：7.42」と「(alcohol,whiskey)：7.27」）、たまたまこの序列で出力した手法が、そうでない手法より高く評価されるという問題がある。

また、単語連想の評価で見たように、counter-fitting は、単語 A の近傍に非関連語の単語 B が比較的多数出てくる。単語(A, B)の単語ペアランキングでの順位は本来低いが、counter-fitting では高くなる。しかしこのような単語(A, B)は SimLex-999 データセットに殆ど無いため、counter-fitting の悪い点が表出しにくいという問題がある。

4.5 言い換え文検索の評価実験

言い換え文検索とは、検索対象文集合の中から、クエリ文の言い換えに相当する文、すなわち、クエリ文と意味的に同一な文を検索するタスクである。変換後概念ベースを言い換え文検索に適用することにより、変換後概念ベースが、変換前概念ベースと比べて、悪い結果をもたらしていないか、より良い結果をもたらしているかを検証した。

言い換え文検索は、1 節でも触れた最もナイーブなロジックで行う。検索対象文とクエリ文それぞれに対し、文中の内容語のベクトルの重心を、該文のベクトルとする。クエリ文ベクトルと各検索対象文ベクトルとの距離の昇順に、検索対象文をランキングする。クエリ文の言い換えに相当する正解文のランキングにおける順位を導出する。この順位が高い程、検索精度が高い。

評価用データとして、[16]のデータを利用した。[16]のデータから 600 個の文をとり、一部を修正したものをクエリ文とした。各クエリ文に対し作成されている、意味的に一致しない文のリストから一部の文を除去ないし修正したも

のをとった。これを、該クエリ文に対応する不正解文と呼ぶことにする。1 クエリ文に対応する不正解文は平均 7.9 個となった。また、各クエリ文に対し、正解文となる言い換え文を 1 個新規に作成した。全クエリ文に対する不正解文と正解文を、重複するものはユニークにしてマージし、検索対象文とした。検索対象文の個数は 5159 となった。

概念ベースごとに、各クエリ文に対し検索対象文をランキングし、該クエリ文に対する正解文の順位を導出し、その平均を出した。

また、クエリ文中のある単語と、対応する不正解文中のある単語とのペアが反義語辞書中にある場合、その不正解文を、反義語を含む不正解文と呼ぶことにする。1 クエリ文に対応する反義語を含む不正解文は平均 1.3 個となった。ランキングにおける、反義語を含む各不正解文の順位も導出し、全クエリ文の反義語を含む全不正解文の導出順位の平均を出した。

以上の順位導出結果を表 5 に示す。

	変換前	counter	retro	提案手法
正解文	114.9	122.6	115.0	111.6
反義語を含む不正解文	48.8	306.2	159.8	165.0

表 5：手法ごとの順位の平均

正解文の順位は、提案手法が、変換前や従来手法よりも高くなった。反義語を含む不正解文の順位は、提案手法は、変換前や retrofitting よりも低くなった。変換前は、正解文より反義語を含む不正解文の方が高順位であったが、変換後は、順位が逆転した。

表 6 に、特定のクエリ文に対する順位を示す。表 6 に見られるように、変換前は、(子供, 大人), (豊か, 貧しい)のような反義語辞書中の反義語ペアのベクトル間距離が近いため、反義語を含む不正解文も、クエリ文とのベクトル間距離が近くなり、高順位となる。変換後は、反義語辞書中の反義語ペアのベクトル間距離が遠くなるため、反義語を含む不正解文も、クエリ文とのベクトル間距離が遠くなり、順位が下がる。提案手法では、その分、正解文の順位が上がっている。

反義語を含む不正解文の順位は、counter-fitting の方が、提案手法よりも低い。counter-fitting の、反義語辞書中の反義語ペアのベクトルを大きく離す性質を反映しているといえる。だが、反義語ペアは、意味的には遠いものの、トピックとしては同じであるため、ある程度は離すべきであるが、離し過ぎるのは問題がある。反義語を含む文も、同様に意味的には遠いものの、トピックとしては同じであるため、ある程度順位は下げるべきだが、下げすぎるのは問題がある。提案手法は、反義語辞書中の反義語ペアのベクトルを適度に離す性質を持つため、この問題を解決しているといえる。

正解文の順位の有意差検定を行ったところ、提案手法と、変換前, counter-fitting, retrofitting それぞれとの間の p 値は、42.7%, 6.8%, 48.2%となり、有意水準 5%で有意差は認められなかった。先述したように、1 クエリ文あたりの反義語を含む不正解文は平均 1.3 個である。反義語を含む不正解文が正解文より順位が下がり、その分、正解文の順位が上がったとしても、正解文の順位は、変換前の正解文の順位である 114.9 から高々1.3 上昇した 113.6 となる。提案手法はそれより高順位となっているが、元々の順位の上がり幅が少ないタスクであるため、有意差が出なかったと考えられる。

また、counter-fitting では、クエリ文中の単語の近傍に非関連語が比較的多数位置するが、そのような非関連語を含む検索対象文があれば、そういった非関連語を含む不正解の検索対象文が上位にあがってくる可能性がある。今回の評価で用いた検索対象文のセットは比較的少数であるため、そのような非関連語を含む検索対象文が殆ど無く悪影響は出なかったが、検索対象文のセットが膨大になるにつれ、クエリ文中の単語の近傍の非関連語を含む不正解の検索対象文が上位に多数出てくる可能性があると考えられる。
提案手法を用いると、変換前と比べ、正解文の順位に有意差はないものの、反義語を含む不正解文の順位を正解文よりも下げることができ、提案手法による単語ベクトルの

●クエリ文:「子供が初めてしゃべった」に対する検索対象文の順位

		word2vec	counter	retro	提案手法
言い換え文	子が始めて発話した	33	113	19	9
反義語を含む不正解文	大人が初めてしゃべった	5	364	26	35

●クエリ文:「豊かな暮らしをする」に対する検索対象文の順位

		word2vec	counter	retro	提案手法
言い換え文	贅沢な生活をする	20	22	16	12
反義語を含む不正解文	貧しい暮らしをする	2	120	4	19
	貧しい生活をする	5	152	12	27

表 6：特定のクエリ文に対する検索対象文の順位

配置がこのタスクに有効であるといえる。

5. まとめ

本稿では、生成済みの概念ベースが与えられたときに、反義語辞書中の反義語ペアのベクトルは適度に離れた位置にあるように、それ以外の単語ペアのベクトル間距離はなるべく変換前の距離を維持するように、概念ベース中の全単語ベクトルを変換する手法を提案した。検証により提案手法は、基点語の近傍における関連語の割合を高くし、単語ベクトル間の差分関係をなるべく維持し、検索の精度を高めることを確認した。提案手法は、基点語の近傍における関連語の割合が高いという特性により、近傍内のノイズとなる単語の存在の悪影響が大きくなるタスクとコンテンツにおいて、特に高い有効性をもつといえる。

提案手法は反対の意味をもつ反義語を対象とした。一方、(東京,大阪)のような対比語も周辺文脈が似通っているため、単語ベクトルが近くなる課題がある。そのため、「東京の交通量は多い」というクエリ文に対し、「大阪の交通量が多い」という検索対象文が上位にくる。対比語を反義語辞書に登録すれば、対比語に対しても提案手法のロジックを適用できる。これにより、対比語のベクトルを離し、上記検索対象文の順位を下げることができる。対比語は、反義語に比して数が膨大である。今後は、対比語を自動獲得する手法の研究を進める予定である。

参考文献

[1] Thomas Hofmann: Probabilistic Latent Semantic Analysis, Proc. UAI'99, pp. 289-296, (1999)

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean: Efficient estimation of word representations in vector space, CoRR, Vol. abs/1301.3781, (2013)

[3] Jeffrey Pennington, Richard Socher, and Christopher D. Manning: GloVe: Global Vectors for Word Representation, EMNLP, Vol. 14, pp. 1532-1543, (2014)

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov: Enriching word vectors with subword information, TACL, pp. 135-146, (2017)

[5] Zellig S. Harris: Distributional structure, Word, Vol. 10, pp. 146-162, (1954)

[6] Masataka Ono, Makoto Miwa, and Yutaka Sasaki: Word Embedding-based Antonym Detection using Thesauri and Distributional Information, NAACL/HLT-2015, pp. 984-989, (2015)

[7] Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu: Learning Semantic Word Embeddings based on Ordinal Knowledge Constraints, ACL, pp. 1501-1511, (2015)

[8] Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith: Retrofitting Word Vectors to Semantic Lexicons, NAACL, http://arxiv.org/abs/1411.4166, (2015)

[9] Robert Speer, and Joanna Lowry-Duda: ConceptNet at SemiEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge, https://arxiv.org/abs/1704.03560, (2017)

[10] Nikola Mrksic, Diarmuid O Seaghdha, Blaise Thomson, Milica Gasic, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, Steve Young: Counter-fitting Word Vectors to Linguistic Constraints, NAACL-HLT, pp. 142-148, (2016)

[11] John Duchi, Elad Hazan, and Yoram Singer: Adaptive subgradient methods for online learning and stochastic optimization, Journal of Machine Learning Research, Vol. 12, pp. 2121-2159, (2011)

[12] Takeshi Fuchi, Shinichiro Takagi: Japanese-Morphological Analyzer using Word Co-occurrence –JTAG-, COLING-ACL, pp. 409-413, (1998)

[13] https://github.com/nmrksic/counter-fitting

[14] Bin Gao, Jiang Bian, and Tie-Yan Liu: WordRep: A benchmark for research on learning word representations, ICML Workshop on Knowledge-Powered Deep Learning for Text Mining, (2014)

[15] https://github.com/kudkudak/word-embeddings-benchmarks/ : WordRep

[16] Yu Takabatake, Hajime Morita, Daisuke Kawahara, Sadao Kurohashi, Ryuichiro Higashinaka, and Yoshihiro Matsuo: Classification and Acquisition of Contradictory Event Pairs using Crowdsourcing, Proc. 3rd Workshop on EVENTS at the NAACL-HLT, pp. 99-107, (2015)