

# 語義・概念ベクトルによる意味タスクの精度向上

金田 健太郎      小林 哲則      林 良彦

早稲田大学理工学術院

kanada@pcl.cs.waseda.ac.jp

## 1 はじめに

近年、大規模なコーパスからニューラルネットワーク、あるいはそれに準じた手法により、単語などの基本的な言語単位の表現 (分散表現) を獲得する手法が注目されており、様々な自然言語処理のタスクへの適用が行われている。なかでも、意味的なタスクに対する適用性の議論が特に盛んである [5]。

Word2Vec[6] に代表される手法により導出される分散表現は、基本的に単語に対するものであり、意味的なタスクへの適用においては、語義、あるいは、概念に対応する分散表現が望まれる。このような意味レベルの分散表現を求める最近の試みの一つとして、AutoExtend と呼ばれる手法 [8] がある。AutoExtend は、Word2Vec などの手法により得られた単語の分散表現を入力とし、WordNet のような構造化された言語知識を用いることにより、語義、概念の分散表現 (以下、語義・概念ベクトル) を導出する。[8] では、語義・概念ベクトルを利用することで、語義解消タスク、文脈ありの単語類似度タスクで従来手法より高いスコアが得られることが報告されている。

本報告では、AutoExtend により導出された語義・概念ベクトルを伝統的な単語間の意味的類似度 (word similarity)、意味的関連度 (word relatedness) といった意味タスク、さらには、近年提示された、単語・語義間の類似度 (word-to-sense similarity) という意味タスクへ適用する手法を提案し、その有効性を標準的なデータセットを用いた評価実験により示す。

## 2 AutoExtend

AutoExtend は、Word2Vec などの手法により得られた単語の分散表現を入力とし、単語の持つ各語義、それぞれの語義が指示する語彙概念に対する分散表現を導出する手法である。この際、テキストコーパスは必要なく、そのかわりに語義・概念の関係を整理した

語彙資源 (WordNet など) を利用する。

AutoExtend の根幹をなす考え方は、単語 (word) はいくつかの語義を持ち (単語と語義の組み合わせが語彙素:lexeme)、これらの語彙素の集合として概念 (synset) が構成される、という WordNet の情報構造に基づく。単語  $w^i$  の概念  $s^j$  に属する語彙素を  $l^{(i,j)}$  としたとき、それぞれを表現するベクトルの関係性は次のように表せる。なお以下では、誤解のない範囲で語彙素のことを語義と書く。

$$\vec{w}^i = \sum_j l^{(i,j)} \vec{s}^j, \quad \vec{s}^j = \sum_i l^{(i,j)}$$

AutoExtend には、単語ベクトルと同じ空間に語義・概念ベクトルを導出できる利点がある。すなわち、各ベクトルの同じ次元は同じ意味合いを共有する。

## 3 本研究における意味タスク

本研究において語義・概念ベクトルを適用する意味タスクと、それぞれにおいて用いるタスクデータセットは以下の3つである。

- 単語間の意味的類似度: 単語の意味的同義性の程度を予測するタスクである。999 個の単語ペアに対して意味的類似度を付与したデータである SimLex-999[3] を用いる。
- 単語間の意味的関連度: 意味的関連性は同義性の程度に基づく類似性よりも広い概念であり、上位・下位関係、全体・部分関係などの様々な意味的關係を含む [1]。353 個の単語ペアについて意味的関連度を付与したデータである WordSim353 [2] を用いる。
- 単語・語義間の類似度: これは、レベルの異なる言語単位間の意味的類似度・関連度を扱う SemEval-2014 Task 3[4] の一部として提案された新しい意味タスクである。基本的には、複数の語義を持つ

単語に対して、指定された語義がどの程度代表的かを判別するタスクであると解釈することができる。AutoExtendにより導出される語義・概念ベクトルは、その入力となる単語ベクトルと同じ空間に与えられたため、この利点を確認するために適した意味タスクである。

## 4 意味的類似度・関連度の計算手法

対象とする意味タスクでは、与えられた単語ペア、あるいは、単語と語義の組に対して、データセット中の gold data に近い類似度・関連度を予測することが求められる。本研究では、大きく2つの手法と、これらを回帰モデルにより統合する手法を提案し、次節で述べる評価実験により比較を行う。

### 4.1 単語の持つ特定の意味に注目した手法

文脈ありの単語類似度タスクとは異なり、いずれの意味タスクにおいても与えられるものは、単語のペア、または、単語と語義の組のみである。このような設定では、被験者は、与えられたペアや組の要素の意味を相互に制約として利用し、類似度・関連度の評定を行うものと想定される。

そこで、特に類似度タスクにおける高い相関を期待し、単語の持つ特定の語義（概念）に注目して類似度・関連度を算出する手法を提案する。2つの単語が与えられた時、それぞれの単語の持つ任意の語義（概念）の組み合わせに対してコサイン類似度を算出し、そのうち最も高かったスコアを与えられた単語ペアの類似度・関連度とする。以下、語義ベクトルを使用した手法を LexemeMax、概念ベクトルを使用した手法を SynsetMax と呼ぶ。単語  $w_1, w_2$  の類似度の算出法は、それぞれの持つ任意の語義を  $l_1^m, l_2^n$ 、その語義が属する概念を  $s_1^m, s_2^n$  としたとき次の式で表せる。

$$\begin{aligned} \text{LexemeMax}(w_1, w_2) &= \max(\text{cossim}(l_1^m, l_2^n)) & l_1^m, l_2^n \\ \text{SynsetMax}(w_1, w_2) &= \max(\text{cossim}(s_1^m, s_2^n)) & s_1^m, s_2^n \end{aligned}$$

ただし Word-2-Sense タスクにおいては単語-語義のペアが与えられているため、単語-語義ペア  $w_1, l_2$  の類似度を次のように算出した。

$$\begin{aligned} \text{LexemeMax}(w_1, l_2) &= \max(\text{cossim}(l_1^m, l_2^n)) & l_1^m \\ \text{SynsetMax}(w_1, l_2) &= \max(\text{cossim}(s_1^m, s_2^n)) & s_1^m \end{aligned}$$

### 4.2 単語の持つ意味全体を考慮した手法

関連度の評定においては、対処とする単語それぞれが持つ様々な意味の組み合わせが総合的に考慮されていることが考えられる。そこで、特に関連度タスクにおける高い相関を期待し、単語の持つ語義（概念）全体を見て類似度・関連度を算出する2手法を提案する。2つの単語が与えられた時、それぞれの単語の持つ語義（概念）のベクトルをすべて足しあわせたものでコサイン類似度を算出し、それを単語ペアの類似度・関連度とする。以下、語義ベクトルを使用した手法を LexemeSum、概念ベクトルを使用した手法を SynsetSum と呼ぶ。式で表すと次のとおりである。

$$\begin{aligned} \text{LexemeSum}(w_1, w_2) &= \text{cossim}\left(\sum_m l_1^m, \sum_n l_2^n\right) \\ \text{SynsetSum}(w_1, w_2) &= \text{cossim}\left(\sum_m s_1^m, \sum_n s_2^n\right) \end{aligned}$$

ただし Word-2-Sense タスクにおいては単語-語義のペアが与えられているため、単語-語義ペア  $w_1, l_2$  の類似度を次のように算出した。

$$\begin{aligned} \text{LexemeSum}(w_1, l_2) &= \text{cossim}\left(\sum_m l_1^m, l_2^n\right) \\ \text{SynsetSum}(w_1, l_2) &= \text{cossim}\left(\sum_m s_1^m, s_2^n\right) \end{aligned}$$

### 4.3 回帰モデルによる類似度・関連度の統合

以上で示した各手法による類似度・関連度を回帰モデルにより統合する。これにより特に、様々な意味関係が非明示的に扱われる意味的関連度のタスクでの精度向上を期待する。また、回帰モデルにおける属性となる各類類似度・関連度の重要度を見ることで、意味タスクに応じたそれぞれの貢献度を調べることができる。なお、ベースラインとして、Word2Vec によるベクトルに対するコサイン類似度を用いる。

統合には RandomForest (RF)、一般線形モデル (Linear)、LassoCV、Ridge、RidgeCV の5手法を用いた。RF モデルが非線形、その他が線形のモデルである。LassoCV 及び RidgeCV は、Lasso モデル、Ridge モデルの最適なパラメータを交差検定により決定する手法であり、scikit-learn<sup>1</sup> に実装されている。

ここでそれぞれのモデルに対し2パターンの属性を用意した。一方では提案する4手法及びベースライン手法によって算出された類似度5つを使用した。もう一方では、それに加えそれぞれの手法でコサイン類似

<sup>1</sup><http://scikit-learn.org/>

表 1: 各手法による結果

手法	SL-999	WS353	W2S
w2v	0.4539	<b>0.6525</b>	0.3707
LexemeSum	0.2331	0.5861	0.2212
LexemeMax	0.3110	0.5941	0.2290
SynsetSum	0.4691	0.5940	<b>0.4066</b>
SynsetMax	<b>0.5270</b>	0.5687	0.3987

表 2: 回帰モデルにより統合した結果

統合手法	SL-999	WS353	W2S
RF	0.2555	0.5156	0.2204
Linear	0.5453	0.6523	0.4153
Ridge	0.5469	<b>0.6642</b>	0.4196
RidgeCV	0.5423	0.6547	0.4106
LassoCV	<b>0.5506</b>	<b>0.6610</b>	<b>0.4259</b>

度算出時に使用したベクトルのペア 5 組を加えた。多次元 (300 次元) のベクトルの組から算出されるコサイン類似度は、それらのベクトルを一次元に圧縮したものと解釈できる。この際に失われた情報を補うために、算出時に使用したベクトルを属性に加えた。以下後者の属性によるモデルは、それぞれのモデル名の後に +v とつけて呼ぶ (例: RF+v)。

## 5 評価実験

前述の 3 種類の評価データそれぞれに対して提案手法により類似度・関連度を算出した後、両者の相関を Pearson の相関係数によって評価した。

今回は、Google の配布する Word2Vec モデル (Continuous Bag-Of-Words に基づく 300 次元のベクトル)<sup>2</sup> と WordNet[7] を用いて作成された Rothe の配布する語義・概念ベクトルを使用した<sup>3</sup>。

### 5.1 主要な結果

それぞれの手法により計算した類似度・関連度と SimLex-999 (SL-999), WordSim353 (WS353), Word-2-Sense (W2S) における gold data との Pearson の相関係数を表 1 から表 3 に示す。ただし、ベクトルが存在せず類似度が算出できなかった単語ペア及び単語-語義ペアは除外している。

回帰モデル使用時には、全評価データに共通して LassoCV+v が最も高いスコアを出している。この回

<sup>2</sup><https://drive.google.com/file/d/0B7XkCwpI5KDYn1\NUTt1SS21pQmM/edit?usp=sharing>

<sup>3</sup><http://www.cis.lmu.de/~sascha/AutoExtend/embeddings.zip>

表 3: 属性にベクトルを追加し回帰モデルにより統合した結果

統合手法	SL-999	WS353	W2S
RF+v	0.1440	0.1298	-0.0080
Linear+v	0.3028	0.2915	0.2459
Ridge+v	0.4158	0.5296	0.2670
RidgeCV+v	0.5355	0.5332	0.2272
LassoCV+v	<b>0.5895</b>	<b>0.6925</b>	<b>0.4323</b>

表 4: LassoCV+v における属性の重要度

属性	SL-999	WS353	W2S
w2v	0.0891	<b>0.4075</b>	0.1935
LexemeSum	0	0.0022	0
LexemeMax	0	0.0226	0
SynsetSum	0	0.019	0.0424
SynsetMax	<b>0.2959</b>	0.1662	<b>0.4032</b>
w1+v	0.1175	0.099	0.1689
w2+v	0.1327	0.1382	0.1701
l1_sum+v	0.0915	0.0845	0.1607
l1_max+v	0.0933	0.0899	0.1746
l2_sum+v	0.1079	0.1117	-
l2_max+v	0.0994	0.1172	-
s1_sum+v	0.0877	0.081	0.1482
s1_max+v	0.0818	0.0784	0.1456
s2_sum+v	0.0929	0.0992	-
s2_max+v	0.087	0.0975	-
l2+v	-	-	0
s2+v	-	-	0.0247

帰モデルにおける一つ一つの属性の重要度は表 4 のようになった。ここで、重要度はモデルにおける属性ごとの回帰係数の絶対値を正規化したものとする。二重線より上の 5 つの属性は類似度・関連度属性であり、それより下の +v が付加されたものは、対応するベクトルを表し、それぞれの重要度は各成分の係数の絶対値の総和としている。

以下それぞれの評価データごとに実験結果の詳細と考察を述べる。

### 5.2 SimLex-999

SimLex-999 は単語間類似度の意味タスクであり、4.1 節で提案した手法が高い相関を示すことが期待されていた。表 1 の結果を見ると、SynsetMax の手法で高いスコアが得られており、上記の期待と一致している。

概念ベクトルを使用した 2 つの手法においてベースラインを超える相関が得られたのに対し、語義ベクトルを使用した 2 つの手法のスコアはベースラインを下回った。語義ベクトルよりも概念ベクトルを使用したほうが高い相関が得られるということは、一般に人が単語の意味について考えるとき、語義的な性質、つまりその意味がどの単語に属しているかについてはあま

り考慮していないということが推測される．表 4 に示された属性の重要度によれば，SynsetMax が最大の重要とを示しており，これも予想と一致している．

### 5.3 Wordsim353

Wordsim353 は単語間関連度の意味タスクであり，提案手法に中では LexemeSum，もしくは，SynsetSum によって高い相関を示すことが期待されていた．しかし表 1 に示した結果をみると，どれも同程度のスコアとなっており，またすべてベースラインを下回っている．評価データが様々な関係性をまとめて関連性として扱っており，語義すべてを一つにまとめて表現している単語ベクトルの方が有効なタスクであったと理解できる．これは表 4 の属性の重要度についても w2v の重要度が最大となっていることからわかる．

またこのタスクにおいては様々な関連性が非明示的に扱われているため，回帰モデルによりスコアを統合した際より良い結果が得られることが期待されていた．表 2 を見ると，期待通りスコアの増加が確認できたが，その増加量は僅かであった．つまり，単語ベクトルが有効なタスクに単語ベクトルから作成された語義・概念のベクトルを併せても，劇的な効果を望むのは難しい事が確認できた．一方で，類似度算出時のベクトルを属性に含めた場合にはスコアの向上が見られた．これは，ベクトルからの情報抽出にあたって，コサイン類似度という単一の指標以上に良い手法が存在する可能性を示唆している．

### 5.4 Word-2-Sense

Word-2-Sense は単語・語義間の類似度に関する意味タスクであり，提案する手法では LexemeMax，もしくは，SynsetMax による類似度が高い相関を示すことが期待されていた．表 1 に示した結果を見ると，単語間意味的類似度のタスクである SimLex-999 と同じく概念ベクトルを使用した 2 つの手法においてベースラインを超える結果が得られ，語義ベクトルを使用した 2 つの手法においてはベースラインを下回る結果となった．しかし，SimLex-999 とは異なり同じベクトルを使用した場合の手法による相関の差が見られなくなっている．タスクが語義と単語という異なるレベルの言語単位を扱っているために，4.1 節で提案した最大値に基づく手法に対する期待が成立しなかったと考えられる．ただし表 4 中の属性の重要度について見てみると，SynsetMax が最大となっており，この手法が

よりよい単語・語義間類似度を表すのに貢献していることがわかる．なお，相関算出に使うデータ数が異なるため厳密な比較は不可能だが，今回試した複数の手法 (SynsetSum，SynsetMax，及びいくつかの回帰モデル使用時) によって，SemEval2014 Task 3 における最高スコアである 0.389 を上回る相関を得た．

## 6 おわりに

AutoExtend 手法により導出された語義・概念ベクトルを利用した意味的類似度・関連度の計算手法を複数提案し，ターゲットとする意味タスクの性質に応じたそれぞれの手法の有効性を評価した．総じて，語義・概念ベクトルは有用であること，特に概念ベクトルに関しては，単体で利用するだけでも類似度タスクにおいて有効であることがわかった．

今後は，次元圧縮による情報の抽出など語義概念ベクトルの有効な利用方法を検討する．また，コサイン類似度だけでなくユークリッド距離も併用した類似度・関連度算出方法の改善を行う．類似度・関連度の統合方法としては，ニューラルネットの適用を試みる．

## 謝辞

本研究は JSPS 科研費 #25280117 の助成を受けた．

## 参考文献

- [1] Budanitsky, A. and Hirst, G. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, Vol.32, No.1, pp.13-47.
- [2] Finkelstein, L., et al. 2001. Placing search in context: The concept revisited. *Proc. of WWW 2001*, pp.406-414.
- [3] Hill, F., et al. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. arXiv preprint arXiv:1408.3456.
- [4] Jurgens, D., et al. 2014. SemEval-2014 Task 3: Cross-level semantic similarity. *Proc. of SemEval 2014*, pp.17-26.
- [5] Levy, O. et al. 2014. Linguistic regularities in sparse and explicit word representations. *Proc. of CoNLL-2014*, pp.171-180.
- [6] Mikolov, T., et al. 2013. Distributed Representations of Words and Phrases and their Compositionality. *Proc. of NIPS 2013*.
- [7] Miller, G.A. 1995. WordNet: a lexical database for English. *Communications of the ACM*, Vol.38, No.11, pp.39-41.
- [8] Rothe, S. and Schütze, H. 2015. AutoExtend: Extending word embeddings to embeddings for synsets and lexemes. *Proc. of ACL 2015*, pp.1793-1803.