単語埋め込みによる論理演算

内藤 雅博 [1,3]　横井 祥 [2,3]　下平 英寿 [1,3]
[1] 京都大学　[2] 東北大学　[3] 理化学研究所
neiteng@sys.i.kyoto-u.ac.jp, yokoi@ecei.tohoku.ac.jp, shimo@i.kyoto-u.ac.jp

1 はじめに

自然言語処理の基盤技術である単語埋め込みは，$\boldsymbol{v}_{king} \approx \boldsymbol{v}_{royal} + \boldsymbol{v}_{man}$ のように埋め込みの加法で意味の合成を近似できるという性質を持つ（**加法構成性**）．

加法構成性が成立する理由を説明するためにいくつかの理論が提示されてきたが [2, 8, 1]，本稿で示す通り，実際の単語埋め込みでは成立しない仮定が置かれている．また，理論が Skip-gram[15] などの特定の手法に依存しており，GloVe[18] などの他の手法との統一した理解は分野に残された課題である．

加法構成性について次のような疑問も残されている：「意味の合成」とひとことで言っても，意味の AND をとりたい場合（例：$king \approx royal \wedge man$；通常の加法構成性に相当）と OR をとりたい場合（例：$case \approx box$（箱）$\vee$ $instance$（事例））では異なる演算が必要だと考えられる．OR の演算は単語埋め込みでどのように計算できるのだろうか？

本稿では，以上の加法構成性に対する疑問に対して以下の理論的・実験的根拠を与える：

1. 単語埋め込みからその平均を引いた（**中心化**した）ベクトルを考えることで Skip-gram with Negative Sampling (SGNS)[16] と GloVe の性質を統一して記述できることを示すとともに，既存の加法構成性の理論の問題点を取り除く (§3).
2. 1. で得た結果を出発点に，表1のような意味の AND・OR・NOT をとりたい場合の埋め込みの計算方法を示す (§4).
3. 本稿の理論の帰結を利用して加法構成性の精度を向上させられること，OR・NOT の演算を正しく行えることを実験で確認する (§5).

2 準備：単語埋め込み

■単語埋め込み手法の紹介 本稿で扱う SGNS や GloVe のような単語間の共起情報による単語埋め込み手法は，情報検索・グラフ埋め込み・推薦システムなど分野をまたいで活躍の幅を広げている [20, 10, 9]．また，近年注目されている BERT[5] なども文脈から単語を予測することで埋め込みを得ており，これらの手法の拡張ともみなせる．

■記号 語彙を V，単語 w の埋め込みを $\boldsymbol{v}_w$，単語 c の文脈側埋め込みを $\boldsymbol{u}_c$ で表す．$p(w,c)$ を単語対 (w,c) が共起する同時確率，$p(c|w)$ を単語 w の文脈が単語 c である確率，$p(w)$ を単語 w の頻度とする．

■単語埋め込みの性質 [14, 21] によれば，最適に学習された SGNS の埋め込みは次を満たす：

$$\log \frac{p(w,c)}{p(w)q(c)} - \log k = \boldsymbol{v}_w^\top \boldsymbol{u}_c. \tag{1}$$

ここで q は負例の分布，k は共起単語対 (w,c) 1 個あたりの負例の個数である．$q = p$ の場合，自己相互情報量 $\mathrm{PMI}(w,c) = \log \frac{p(w,c)}{p(w)p(c)}$ から $\log k$ を引いた行列を分解していると解釈できる．一方 GloVe では共起行列を直接分解するアプローチをとっており，最適に学習された埋め込みは次を満たす：

$$\log p(w,c) = \boldsymbol{v}_w^\top \boldsymbol{u}_c + a_w + b_c - \log Z. \tag{2}$$

ここで a_w, b_c はバイアス項，Z は正規化定数である．以下では，SGNS・GloVe の各埋め込みは (1), (2) を満たすと仮定する．

3 SGNS と GloVe に共通する構造

既存の加法構成性の理論 [1] では

$$\mathrm{PMI}(w,c) = \boldsymbol{v}_w^\top \boldsymbol{u}_c \tag{3}$$

という仮定を置いていたが，本稿の実験 (§5.1) が示す通り実際には成り立っていない．また，(3) は §4以降の OR・NOT の理論でも重要な役割を果たす．本節では，SGNS・GloVe の単語埋め込みを頻度で重み付けて中心化するだけで，仮定 (3) が成り立つように補正できることを示す．§5.2ではこの補正を行うことで加法構成性の精度が向上することを確認しており，各下流タスクへの応用が期待できる．

はじめに，SGNS・GloVe の埋め込みを中心化することで (3) が誤差項を含む形で導かれることを示す．

表 1: AND，OR，NOT の演算の説明.

AND の演算	（通常の加法構成性）	意味の「かつ」をとる演算	$king \approx royal \wedge man$
OR の演算	（多義語の構成）	意味の「または」をとる演算	$case \approx box$（箱）$\vee$ $instance$（事例）
NOT の演算	（反義語の構成）	意味の「否定」をとる演算	$hate \approx \neg love$

平均を $\bar{\boldsymbol{v}} = \sum_w p(w)\boldsymbol{v}_w$，$\bar{\boldsymbol{u}} = \sum_c p(c)\boldsymbol{u}_c$，中心化した単語埋め込みを $\tilde{\boldsymbol{v}}_w = \boldsymbol{v}_w - \bar{\boldsymbol{v}}, \tilde{\boldsymbol{u}}_c = \boldsymbol{u}_c - \bar{\boldsymbol{u}}$ とおく．

定理 1. SGNS・GloVe の埋め込みがそれぞれ (1),(2) を満たす時，次の等式を満たす：

$$\mathrm{PMI}(w,c) = \tilde{\boldsymbol{v}}_w^\top \tilde{\boldsymbol{u}}_c + \bar{\epsilon} - \epsilon_w - \epsilon_c. \tag{4}$$

ただし $\bar{\epsilon}, \epsilon_w$ は誤差項で，$\epsilon_w = D_{\mathrm{KL}}(p(\cdot)\|p(\cdot|w)), \bar{\epsilon} = \sum_w p(w)\epsilon_w$．（証明: 付録 A.1）

次の命題により，$|\mathrm{PMI}(w,c)| \ll 1$ である時に誤差項が無視できることが分かる．

命題 2. $\Delta = \max_{w,c} |\mathrm{PMI}(w,c)|$ とおくと，$\epsilon_w = O(\Delta^2), \bar{\epsilon} = O(\Delta^2)$, $(\Delta \to 0)$ である．（証明:付録 A.2）

実際のデータでは $|\mathrm{PMI}(w,c)| \ll 1$ は成り立っていないが，後述の実験により，中心化によって仮説 (3) の誤差が大幅に改善することを確認した (§5.1).

4 単語埋め込みによる論理演算

4.1 AND の演算

[1] は，§3で近似的に得られた PMI 分解の構造 (3) が厳密に満たされる時，$king = royal \wedge man$ のような AND の意味演算がベクトルの加法

$$\boldsymbol{v}_{king} = \boldsymbol{v}_{royal} + \boldsymbol{v}_{man} \tag{5}$$

に対応することを示した（**AND の公式**；通常の加法構成性）．証明の概略は次の通りである：各文脈 c での $king$ の出現確率が $p(king|c) = p(royal|c)p(man|c)$ のような積で表される考え，PMI の log を介して確率の積を埋め込みの足し算に対応させる．

4.2 OR の演算

§1で触れたように，AND の演算の他に「または」の演算も考えられる．本節では，OR の演算が埋め込みの頻度重み付き和で実現できることを示す．

■ OR の意味を持つ単語のモデリング 語義 $\{w_i\}_{i=1}^s$ を持つ多義語を w とする．ある単語 w' を固定した時，その文脈として w が出現する確率は w_i が出現する確率の和 $(i = 1, \ldots, s)$ であると考えられる：

$$\forall w' \in V, \quad p(w|w') = \sum_{i=1}^{s} p(w_i|w'). \tag{6}$$

(6) よりただちに $p(w) = \sum_{i=1}^s p(w_i)$ を得る．

■ OR の公式の導出

定理 3 (OR の公式 [13])**.** 単語 $w_1, \ldots, w_s$ と，それらの OR の意味を持つ単語 w の埋め込みが OR のモデリング (6) と PMI 分解の関係式 (3) を満たすと仮定する．$|\mathrm{PMI}(w,c)| \ll 1$ であるとき，OR の公式

$$\boldsymbol{v}_w \approx \sum_{i=1}^{s} \frac{p(w_i)}{p(w)} \boldsymbol{v}_{w_i} \tag{7}$$

が成り立つ．（証明:付録 A.3）

実際のデータでは $|\mathrm{PMI}(w,c)| \ll 1$ は成り立っていないが，後述の実験で OR の公式が良く成り立つことを確認した (§5.3).

4.3 条件付き埋め込みと NOT の演算

仮定 (3) を用いると，AND・OR のみならず NOT の演算も実現できる．本節では，局所的な単語間の関係を表す単語埋め込みである**条件付き埋め込み**の概念を用いて NOT の演算を定式化し，反義語が**マイナス**で表現できることを示唆する．

■ NOT の意味を持つ単語のモデリング

人間の感覚に反し，反義語は違っていると同時に似ているという性質を持つ [4, 22]．たとえば $hate$ と $love$ は反対の意味を持つが，どちらも感情を表すという点で同じである．そのため反義語は似た文脈で出現しやすく，昨今主流である分布仮説 [12, 7] に基づく単語埋め込みでは類似度が高くなる[1)]．このように類義語と反義語の区別は難しく，反義語がどのように埋め込まれているかの理解を困難にしている．本節ではこの反義語の類似性を考慮した定式化を行うことで，反義語の謎を紐解く．

次のような例を考える：$mother$ の反対は，「親」のくくりの中では $father$ だが，「親子関係」のくくりでは $daughter$ である．このように反義語を考える際はこの「くくり」が所与である必要があり，本稿ではこれを単語の集合 A で表現する[2)]．小さな単語集合 A の中で見たとき，反義語 $\neg w$ は単語 w の逆の文脈で出現するであろうという直感から，NOT の意味を持つ単語の共起確率は次のように定式化できる：

$$p(\neg w|\neg w \in A, w') = p(\neg w \in A \backslash \{w\}|\neg w \in A, w') \tag{8}$$

1) SGNS での $hate$ と $love$ の cos 類似度は 0.5 程度である
2) これは反義語の類似性の部分に相当する．

(8) の確率には条件づけに $w \in A$ が登場するため，通常の単語埋め込みではなく，A で出現確率を条件づけた埋め込みが必要になる．本稿では，この埋め込みを A **上の条件付き埋め込み**と呼ぶ．

■条件付き埋め込み

(14) の類推から，条件付き埋め込み $\boldsymbol{v}_{w|A}$ の満たすべき等式は次のように定式化できる：

$$p(w|w \in A, w') = p(w|w \in A) \exp(\boldsymbol{v}_{w|A}^\top \boldsymbol{u}_{w'}) \quad (9)$$

OR の公式 (7) を厳密に認めて左辺を計算すると：

$$p(w|w \in A, w') = \frac{p(w, w \in A|w')}{p(w \in A|w')} = \frac{p(w)\exp(\boldsymbol{v}_w^\top \boldsymbol{u}_{w'})}{p(A)\exp(\boldsymbol{v}_A^\top \boldsymbol{u}_{w'})}$$
$$= p(w|w \in A)\exp((\boldsymbol{v}_w - \boldsymbol{v}_A)^\top \boldsymbol{u}_{w'}).$$

ここで $p(A) = \sum_{w \in A} p(w), \boldsymbol{v}_A = \sum_{w \in A} \frac{p(w)}{p(A)} \boldsymbol{v}_w$．よって条件付き埋め込みは $\boldsymbol{v}_{w|A} = \boldsymbol{v}_w - \boldsymbol{v}_A$ と計算できることがわかる．これにより，一般に行われるある単語集合での中心化（例：国名-首都 の単語埋め込みを PCA で可視化する際の暗黙的な中心化）は，「集合内の単語間の相互関係の精緻化」と説明できる．

補足：大まかに言うと，条件付き埋め込みはコーパスを A に絞った単語埋め込みと理解できる．

■ NOT の公式の導出 OR の公式を厳密に認めて (8) の右辺をさらに計算すると，次を得る：

$$p(\neg w \in A\backslash\{w\}|\neg w \in A, w') = \frac{p(\neg w \in A\backslash\{w\}|w')}{p(\neg w \in A|w')}$$
$$= \frac{p(A\backslash\{w\})}{p(A)} \exp((\boldsymbol{v}_{A\backslash\{w\}} - \boldsymbol{v}_A)^\top \boldsymbol{u}_{w'}). \quad (10)$$

よって $\boldsymbol{v}_{\neg w|A} = \boldsymbol{v}_{A\backslash\{w\}} - \boldsymbol{v}_A$ であり，さらに計算すると次を得る（**NOT の公式**）．

$$\boldsymbol{v}_{\neg w|A} = \boldsymbol{v}_{A\backslash\{w\}} - \boldsymbol{v}_A = -\frac{p(w|w \in A)}{1 - p(w|w \in A)} \boldsymbol{v}_{w|A}. \quad (11)$$

(11) より，条件付けることによって NOT の意味を持つ単語の埋め込みが**マイナス**で計算できることがわかる．(8) は厳密には w の反義語というよりは A の中での w の補集合に当たる単語をモデリングしているが，(11) より反義語の埋め込みがマイナスの方向に存在することが示唆される．

5 実験

5.1 中心化と PMI 分解

PMI 分解の関係式 $\mathrm{PMI}(w, c) = \boldsymbol{v}_w^\top \boldsymbol{u}_c$ の精度が中心化によって向上することを確かめるために，誤差 $e_{wc} = \mathrm{PMI}(w, c) - \boldsymbol{v}_w^\top \boldsymbol{u}_c$ の分布をプロットした（図 1）．orig は元の埋め込み，freq は §3 で議論された

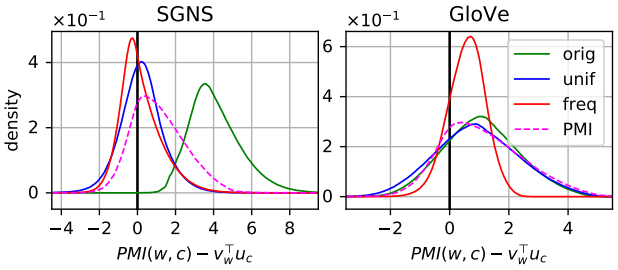

図 1: 1 回以上共起した単語対 (w, c) に対し誤差 e_{wc} の分布をプロットした．誤差の大きさの参考に，$\mathrm{PMI}(w, c)$ 自体の分布を紫破線で示してある．

頻度重み中心化 $\boldsymbol{v}_w \leftarrow \boldsymbol{v}_w - \bar{\boldsymbol{v}}$ を行ったときの結果である．さらに比較として，単語埋め込みの後処理として一般的に行われる [17] $\boldsymbol{v}_w \leftarrow \boldsymbol{v}_w - \sum_{w'} \boldsymbol{v}_{w'}/|V|$（一様中心化）での結果を unif で示している．

結果から，普通の中心化ではなく，特に**頻度重み**で中心化することによって PMI 分解の構造が強くなることがわかる．

5.2 AND の公式の精度評価

AND の公式 (5) の確認を行う．公式の精度が PMI 分解の構造 (3) の有無によってどう変わるのかを観察するために，orig，unif，freq の設定を試した．

評価のために，AND の意味を持つ単語を含むコーパスを次のように作成し，埋め込みを学習した：

複合語のデータセット [6] に含まれる各複合語について，コーパス上の複合語を 1 個の単語に置換する（例: `card game` → `card_game`）．

評価指標としては，データセット [6] に含まれる各複合語 `word1 word2` について，(5) で計算されたベクトルと cos 類似度の高い単語を検索した時に `word1_word2` が何番目に出てくるか（**予測順位**）を計算した．予測順位の累積度数分布を図2 に，要約統計量として平均逆順位を表2に示す．

結果を見ると，特に**頻度重み**で中心化することで予測が正確になっているため，PMI 分解の構造が加法構成性の成立において重要であると考えられる．

5.3 OR の公式の精度評価

OR の公式 (7) が成り立つことを確かめる．

次の手順で架空の多義語を 500 個作成し，その埋め込みを学習した：

1. ランダムに選んだ単語対から架空の多義語を作成し（例：`apple`, `car` →`apple_OR_car`），元の単語を全て架空の多義語に置き換えることで新し

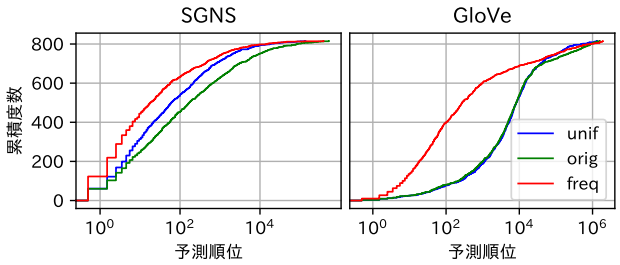

図 2: AND の公式による予測順位の累積度数分布．線が左上にあればあるほど良い．

表 2: AND の公式による予測順位の平均逆順位．

	orig	unif	freq
SGNS	0.15	0.17	**0.28**
GloVe	0.014	0.014	**0.061**

いコーパス を作成した．

2. 元のコーパスと 新しいコーパスを結合したコーパスで単語埋め込みを学習した．

§5.2 と同様に予測順位で評価する．予測順位の平均は SGNS で **1.012**, GloVe で **1.000** であった．OR の公式は近似式であったが，ほぼ 1 番目に予測できるほど公式の精度が高いことが分かる．

5.4 条件付き埋め込みと NOT の実験

-9 から 9 までの数字の埋め込みの可視化を図3に示す．1 と 9 は正の数 $A = \{1, \ldots, 9\}$ 上の条件付き埋め込みでは原点[3)]を挟んでマイナスの方向に位置しており，NOT の公式 (11) を裏付ける．一方で，条件づける集合 A を負の数まで含めた集合に広げると，一転して 1 と 9 は原点から見て似た方向に位置する．この場合，正の数と負の数が原点をはさんで反対側の方向にあるという意味で NOT の公式を支持している．このように，反義語はどのくくりの中で反対と考えるのかが重要であり，NOT の公式がこの事実をうまく定式化できていることが分かる．

6 先行研究との関連

加法構成性については次の先行研究がある：

- [2, 3] は文脈に関する潜在変数モデルを考えることで，Analogy と OR の演算を説明した．このモデルは文脈ベクトルがランダムウォークすると仮定しているが，本稿の理論では必要としない．

3) 原点は頻度を一定と考えて計算している．集合内の単語の頻度が一定になるように，たとえば頻度の高い単語をコーパスからランダムに削除しても $\mathrm{PMI}(w, c) = \boldsymbol{v}_w^\top \boldsymbol{u}_c$ の左辺は変わらないため，頻度を一定と考えても本質的に問題はない．

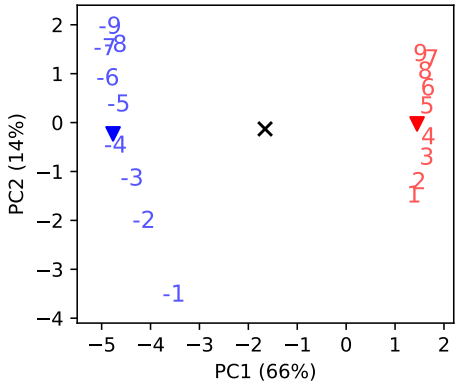

図 3: 数字の埋め込みを PCA を用いて可視化した．× は $A = \{-9, \ldots, 9\} \backslash \{0\}$ としたときの条件付き埋め込み $\boldsymbol{v}_{w|A}$ の原点，▼, ▼はそれぞれ A を正の数，負の数に絞ったときの原点である．

- [8] は，Skip-gram モデル [15] に仮定 $p(w) = 1/|V|$ を置いて AND の演算を説明した．しかし，実際には単語の頻度は非常に偏った分布をしているおり [19]，仮定の妥当性には疑問が残る．本研究ではこの仮定を必要としていない．
- [1] は，仮定 (3) を出発点に AND の演算と Analogy を説明した．しかしながら，著者も認めている通り，GloVe ではバイアス項が存在するため学習結果に任意性があり仮定は正確には成り立っていない．また，SGNS では AND の公式に $\log k$ の分だけずれが入る．本研究の貢献の 1 つは，§3の議論によりこの問題を解決したことである．
- [13] は SGNS で AND と OR の演算を説明した．本稿は，考察対象に GloVe を追加し，NOT の演算の理論を付け加えたものである．

7 おわりに

本稿では，中心化という簡単な操作を介すだけで SGNS と GloVe が共通の構造を持つこと，加法構成性の正確な成立のために中心化が必要であることを示した．また，AND に加え OR, NOT の埋め込みをどのように計算すればよいかを示した．

今後の展望としては，BERT などの文脈を考慮した単語埋め込みでも同様のことが成り立つかどうか，理論・実験の両面から洞察を深めていきたい．

◆**謝辞** 本研究をはじめるにあたってソースコードを提供していただいた Geewook Kim 氏に深く感謝します．本研究は，JST，ACT-X，JPMJAX200S の支援を受けたものです．本研究は JSPS 科研費 20H04148 の助成を受けたものです．

参考文献

[1]Carl Allen and Timothy Hospedales. Analogies explained: Towards understanding word embeddings. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 223–231, 2019.

[2]Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 385–399, 2016.

[3]Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear Algebraic Structure of Word Senses, with Applications to Polysemy. *Transactions of the Association for Computational Linguistics*, Vol. 6, pp. 483–495, 2018.

[4]David A. Cruse. *Lexical Semantics*. Cambridge University Press, Cambridge, UK, 1986.

[5]Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

[6]Meghdad Farahmand, Aaron Smith, and Joakim Nivre. A Multiword Expression Data Set: Annotating Non-Compositionality and Conventionalization for English Noun Compounds. In *Proceedings of the 11th Workshop on Multiword Expressions*, pp. 29–33, 2015.

[7]J. R. Firth. A synopsis of linguistic theory 1930-55. Vol. 1952-59, pp. 1–32, 1957.

[8]Alex Gittens, Dimitris Achlioptas, and Michael W. Mahoney. Skip-gram - Zipf + Uniform = Vector Additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 69–76, 2017.

[9]Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. E-Commerce in Your Inbox: Product Recommendations at Scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 1809–1818, 2015.

[10]Aditya Grover and Jure Leskovec. Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 855–864, 2016.

[11]Michael U. Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *The Journal of Machine Learning Research*, Vol. 13, pp. 307–361, 2012.

[12]Zellig Harris. Distributional structure. *Word*, Vol. 10, No. 2-3, pp. 146–162, 1954.

[13]Geewook Kim, Sho Yokoi, and Hidetoshi Shimodaira. 単語埋め込みの二種類の加法構成性. 言語処理学会 第 26 回年次大会 発表論文集, pp. 724–727, 2020.

[14]Omer Levy and Yoav Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems*, pp. 2177–2185, 2014.

[15]Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR, Workshop Track Proceedings*, 2013.

[16]Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

[17]Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*, 2018.

[18]Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.

[19]Steven Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin and review*, Vol. 21, , 2014.

[20]Dwaipayan Roy, Debasis Ganguly, Sumit Bhatia, Srikanta Bedathur, and Mandar Mitra. Using Word Embeddings for Information Retrieval: How Collection and Term Normalization Choices Affect Performance. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, p. 1835–1838, 2018.

[21]Jun Suzuki and Masaaki Nagata. A Unified Learning Framework of Skip-Grams and Global Vectors. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 186–191, 2015.

[22]Caroline Willners. *Antonyms in Context : A Corpus-Based Semantic Analysis of Swedish Descriptive Adjectives*. PhD thesis.

付録

A 証明

A.1 定理 1の証明

証明 SGNS の場合 $\zeta_w = 0, \xi_c = \log \frac{q(c)}{p(c)}, \gamma = \log k$，GloVe の場合 $\zeta_w = a_w - \log p(w), \xi_c = b_c - \log p(c), \gamma = -\log Z$ と置くと，

$$\mathrm{PMI}(w,c) = \boldsymbol{v}_w^\top \boldsymbol{u}_c + \zeta_w + \xi_c + \gamma \tag{12}$$

を満たす．(12) の両辺に $p(w)$ を掛けて $w \in V$ に関して和をとると

$$-\epsilon_c = \bar{\boldsymbol{v}}^\top \boldsymbol{u}_c + \bar{\zeta} + \xi_c + \gamma. \tag{13}$$

ここで $\bar{\zeta} = \sum_w p(w)\zeta_w$．(12) と (13) より

$$\mathrm{PMI}(w,c) = \tilde{\boldsymbol{v}}_w^\top \boldsymbol{u}_c + (\zeta_w - \bar{\zeta}) - \epsilon_c. \tag{14}$$

(14) の両辺に $p(c)$ を掛けて $c \in V$ に関して和をとると

$$-\epsilon_w = \tilde{\boldsymbol{v}}_w^\top \bar{\boldsymbol{u}} + (\zeta_w - \bar{\zeta}) - \bar{\epsilon}. \tag{15}$$

(14) と (15) より

$$\mathrm{PMI}(w,c) = \tilde{\boldsymbol{v}}_w^\top \tilde{\boldsymbol{u}}_c + \bar{\epsilon} - \epsilon_w - \epsilon_c.$$

□

■中心化しないことによる誤差 SGNS の場合，(3) の誤差は $\xi_c + \gamma = \log \frac{kq(c)}{p(c)}$ である．GloVe の場合は誤差が $\zeta_w + \xi_c + \gamma = a_w - \log p(w) + b_c - \log p(c) - \log Z$ だけ生じる．バイアス項 a_w, b_c の学習には任意性があるため[4]理論的に誤差が大きいことを言うのは難しいが，実験的に誤差が大きいことを確認している (§5.1).

■ Skip-gram についての補足 本定理は Skip-gram $p(c|w) \propto \exp(\boldsymbol{v}_w^\top \boldsymbol{u}_c)$ でも同様に成立する．したがって AND・OR・NOT の理論は Skip-gram にも適用できるが，計算量の問題から実際には SGNS が利用されるため本文中では言及しなかった．SGNS は Skip-gram の近似と説明されがちだが，SGNS は Skip-gram に Noise Contrastive Estimation[11] を適用してからさらに式を簡単化しているため，実際にはモデリング自体も異なる．よって，理論の上では両者は同一視できず，SGNS と Skip-gram を中心化によって同時に議論できることは非自明な結果であることに注意されたい．

A.2 命題 2の証明

証明 Δ の定義から $(w,c) \in V^2$ に一様に $-1 + \frac{p(w,c)}{p(w)p(c)} = -1 + \exp(\mathrm{PMI}(w,c)) = O(\Delta)$ である．

誤差 ϵ_w のオーダーは，$w \in V$ に一様に

$$\begin{aligned}
\epsilon_w &= -\sum_c p(c) \log \frac{p(w,c)}{p(w)p(c)} \\
&= -\sum_c p(c) \left[\left(-1 + \frac{p(w,c)}{p(w)p(c)} \right) + O\left(\left| -1 + \frac{p(w,c)}{p(w)p(c)} \right|^2 \right) \right] \\
&= -\sum_c p(c) O\left(\left| -1 + \frac{p(w,c)}{p(w)p(c)} \right|^2 \right) = O(\Delta^2).
\end{aligned}$$

ただちに $\bar{\epsilon} = O(\Delta^2)$ も従う． □

A.3 定理 3の証明

証明 (6) の両辺を計算する：

$$\begin{aligned}
p(w|w') &= p(w)\exp(\mathrm{PMI}(w,w')) \\
&\approx p(w)\,(1 + \mathrm{PMI}(w,w')) \\
&= p(w)(1 + \boldsymbol{v}_w^\top \boldsymbol{u}_{w'}), \qquad (16) \\
\sum_{i=1}^s p(w_i|w') &= \sum_{i=1}^s p(w_i)\exp(\mathrm{PMI}(w_i,w')) \\
&\approx \sum_{i=1}^s p(w_i)(1 + \boldsymbol{v}_{w_i}^\top \boldsymbol{u}_{w'}) \\
&= p(w)\left[1 + \left(\sum_{i=1}^s \frac{p(w_i)}{p(w)} \boldsymbol{v}_{w_i} \right)^\top \boldsymbol{u}_{w'} \right]. \qquad (17)
\end{aligned}$$

任意の $w' \in V$ に対し (16) ≈ (17) が成り立つことから (7) が従う． □

B 実験設定の補足

特筆した事項以外はすべて実装[5]のデフォルトパラメータを用いた．

B.1 §5.1の詳細

■コーパス 低頻度語（出現回数 < 100）を削除した text8 コーパス[6] を用いた．**■単語埋め込みの学習設定** 埋め込みの次元は 300 次元，窓の大きさは左右 5 単語以内（単語間の距離による重み付けなし）として 100 iteration 学習を行った．SGNS では負例数は $k = 15$ とし，高頻度語の確率的除去は無効にした．GloVe の重み付き最小 2 乗法のパラメータは $x_{\max} = 100$ を用いた．**■その他** freq, unif は $\boldsymbol{u}_j$ 側も中心化を行っている．

B.2 §5.2，§5.3の詳細

■コーパス Wikipedia[7] (2.1G tokens) を用いた．**■単語埋め込みの学習設定** 埋め込みの次元は 300 次元，窓の大きさは左右 5 単語以内とした．SGNS の負例数は $k = 15$，GloVe の重み付き最小 2 乗法のパラメータは $x_{\max} = 100$ を用いた．**■ §5.2の詳細** 複合語のデータセット [6] では複合語の意味がそれを構成する単語の意味の合成になっているかどうかで 5 段階のスコアがついている．スコアが低いと AND の意味を持つといえないため，スコアが 4 か 5 の複合語のみ実験に用いた．**■ §5.3の詳細** 架空の多義語を作る際，元となる単語は出現回数が 100 回以上ものみ用いた．**■その他** 予測順位の計算では検索の際 `word1` と `word2` は除いた．

B.3 §5.4の詳細

■単語埋め込み Common Crawl コーパス (840G tokens) で訓練済みの GloVe を用いた[8]．**■その他** 0 は正負が定義できないため実験からのぞいている．

4) 各バイアス項は，内積 $\boldsymbol{v}_w^\top \boldsymbol{u}_c$ で表現可能である．

5) https://github.com/tmikolov/word2vec, https://github.com/stanfordnlp/GloVe

6) http://mattmahoney.net/dc/textdata.html

7) https://dumps.wikimedia.org/

8) https://nlp.stanford.edu/projects/glove/