文脈限定Skip-gramによる同義語獲得に関する研究

城光 英彰　　松田 源立　　山口 和紀
東京大学 総合文化研究科
{hideaki, matsuda, yamaguch}@graco.c.u-tokyo.ac.jp

1 はじめに

自然言語処理において高度な意味処理を実現する上で，同義語の自動獲得・自動判定は重要な課題である [6][5]．同義語自動獲得・自動判定については様々な手法が提案されているが (例: [5][7][9])，本研究では，同義語獲得において「同じ文脈に現れる単語は類似した意味を持つ」という分布仮説 (distributional hypothesis)[4] や，実際に文脈情報が同義語判定に有用であるとの報告 [2] に基づき，文脈情報を活用するアプローチを検討する．文脈情報の獲得にも手法が多数存在するが，近年では，分布仮説に基づきニューラルネットワーク的な手法を用いて単語の”意味”を表すベクトル (単語ベクトル) を求める Skip-gram モデル [3] が注目されている．Skip-gram モデルで得られた単語ベクトルを利用するとコサイン類似度により単語の意味の類似度が計算できることが知られている．しかし，Skip-gram モデルでは周辺単語の品詞や語順を無視したものを文脈情報として用いており，有用な情報を無視している可能性がある．実際に既存の Skip-gram モデルでは同義語判定に失敗する例として，「カタカナ語」と「和語/漢語」からなる同義語対の場合，コサイン類似度が低くなることなどが知られており [8]，改善が望まれる．

そこで，本研究では， Skip-gram を拡張し，周辺単語の品詞情報や語順情報を取り込み可能なモデル (文脈限定 Skip-gram) を提案する．文脈限定 Skip-gram では，従来の Skip-gram と違い，周辺の単語のうち，ある条件を満たすもの (特定の単語分類属性 (品詞等) や特定の相対位置) のみを文脈として利用し，単語ベクトルを学習する．たとえば,「カタカナ語」あるいは「非カタカナ語」のみに周辺単語を限定することによって，周辺の「カタカナ語」との関係を強く反映した単語ベクトルを学習することができる．そして，そのような様々な限定条件ごとに単語ベクトル及びコサイン類似度を計算し，それらを線形 SVM にて合成することで，同義語判定を行った．その結果，従来の Skip-gram に比べて判定性能を大幅に向上させることができた．

本論文の構成は以下のとおりである．第 2 節では，提案手法について述べる．2.1 節では，従来の Skip-gram モデルについて概説する．2.2 節では，提案する文脈限定 Skip-gram モデルについて説明する．第 3 節では実験結果について述べる．3.1 節では実験に使用したコーパス及び同義語対/非同義語対の教師データ作成方法について述べる．3.2 節では，提案手法による結果を示し，有効性を議論する．最後に第 4 節において結論を述べる．

2 提案手法

2.1 従来の Skip-gram モデル

ここでは Skip-gram モデル [3] について概説する．Skip-gram モデルは，ニューラルネットワーク的な手法を用いて，コーパスの文脈情報から，各単語の単語ベクトルを学習する手法の一種である．Skip-gram モデルでは，ある単語 w_t が文章内の位置 t に存在した場合，その周辺単語 $w_{t+j}(j \neq 0)$ の発生確率 $p(w_{t+j}|w_t)$ を以下の式で与える．

$$p(w_{t+j}|w_t) \propto e^{v'(w_{t+j})^T v(w_t)} \quad (1)$$

ここで，ニューラルネットワークモデル的に言えば，$v(w)$ はある入力単語 (中心単語)w に依存した入力用ベクトル，$v'(w)$ はある周辺単語 w の出力確率を計算するための出力用ベクトルである．v および v' の次元は事前に与えられる．出力確率は，入力用ベクトルと出力用ベクトルの内積に依存し，内積が大きい程確率は高くなる．本論文では，わかりやすさのため，$v(w)$ を単語 w の単語ベクトル，$v'(w)$ を文脈ベクトルと呼ぶことにする，なお，確率分布は 1 に正規化されるので，語彙に含まれるすべての単語 w での正規化によ

り，$p(w_{t+j}|w_t)$ は以下で与えられる．

$$p(w_{t+j} \mid w_t) = \frac{e^{v'(w_{t+j})^T v(w_t)}}{\sum_w e^{v'(w)^T v(w_t)}} \quad (2)$$

さらに $p(w_{t+j}|w_t)$ から，あるコーパスが与えられたときの尤度関数 ℓ を以下の式 (3) で定義する．

$$\ell = \sum_{t=1}^{T} \sum_{-c \le j \le c, j \ne 0} \log p(w_{t+j} \mid w_t) \quad (3)$$

ここで T はコーパスのサイズ，c は文脈窓サイズであり，$1 \le c \le K$ の範囲で一様分布でランダムに決定される．K は事前に与えられる最大文脈窓サイズである．実際のコーパスを利用して，ℓ を最大化する単語ベクトル $v(w)$ および文脈ベクトル $v'(w)$ を求めることが，Skip-gram モデルにおける学習である．なお，本来のモデルは以上の通りであるが，尤度関数 ℓ をこのままの形で最大化することは，計算量等の問題で困難であるため，実際にはいくつかの近似が用いられる．例えば，[3] では，階層的 softmax モデル近似が利用されているが，本論文では説明を省略する．

2.2 文脈限定 Skip-gram モデル

従来の Skip-gram モデルでは，周辺単語として，文脈窓の中に存在するすべての単語を利用している．そのため，文脈単語の種類，語順等の情報を利用することはできない．本研究では文脈として利用される単語を限定することで，Skip-gram を改良する．なお，単語ベクトルの推定に文脈での語順を考慮した既存研究として，[10] があるが，本研究ではより一般的な枠組みを構築する．

文脈限定 Skip-gram モデルでは，式 3 の目的尤度関数 ℓ が以下のように変更される．

$$\ell = \sum_{t=1}^{T} \sum_{-c \le j \le c, j \ne 0} \log p(w_{t+j} \mid w_t) \phi(w_{t+j}, j) \quad (4)$$

ここで，文脈限定関数 $\phi(w_{t+j}, j)$ は，周辺単語 w_{t+j} および相対位置 j がある条件を満たす時のみ 1 となり，それ以外は 0 となる関数である．詳細は省略するが，式 4 は従来の Skip-gram と同様の方法で最大化することが可能である．なお，w_{t+j} と j に関係なく常に 1 となる文脈限定関数 (ϕ_{ALL} と呼ぶ) においては，式 4 は式 3 と同一である．さて，本研究では，基本的な文脈限定関数 $\phi(w_{t+j}, j)$ として，周辺単語の品詞，種類に依存した $\phi^p_{\mathrm{POS}}(w_{t+j})$，周辺単語の左右に依存した $\phi^p_{\mathrm{LR}}(j)$，周辺単語の相対距離に依存した $\phi^p_{\mathrm{WO}}(j)$ の 3 種類を用いる．さらに，それらの組み合わせとして「POS-LR」「POS-WO」も利用する．

表 1: 文脈限定関数の個数一覧

ALL	POS	LR	WO	POS-LR	POS-WO
1	11	2	20	22	220

$\phi^p_{\mathrm{POS}}(w)$ は，単語 w が品詞等のある分類属性を持つ時のみ 1 となる文脈限定関数である．本論文では，「副詞，助詞，動詞，名詞，固有名詞，形容詞，接頭詞，数，記号，カタカナ，非カタカナ」の計 11 個を分類属性として利用する．従って，$\phi^1_{\mathrm{POS}}(w), \ldots, \phi^{11}_{\mathrm{POS}}(w)$ の 11 種類が存在する．$\phi^p_{\mathrm{LR}}(j)$ は，j が正の時のみ 1 となる関数もしくは j が負の時のみ 1 となる関数である．言い換えれば，周辺単語が右側にある場合と左側にある場合に対応しており，2 種類存在する．$\phi^p_{\mathrm{WO}}(j)$ は，$\phi^p_{\mathrm{LR}}(j)$ のある種の拡張であり，$j = p$ の時のみ 1 となる関数である．p は $p = -10, \ldots, -1, 1, \ldots, 10$ として与えられ，文脈窓の特定の相対位置にある時のみに限定する 20 種類の関数となる．さらに，組み合わせにより，$\phi^{pq}_{\mathrm{POS-LR}}(w, j) = \phi^p_{\mathrm{POS}}(w)\,\phi^q_{\mathrm{LR}}(j)$ および $\phi^{pq}_{\mathrm{POS-WO}}(w, j) = \phi^p_{\mathrm{POS}}(w)\,\phi^q_{\mathrm{WO}}(j)$ として新たな文脈限定関数を構成可能である．表 1 に構成可能な文脈限定関数の個数一覧を示す．一つの文脈限定関数に関して一つの Skip-gram モデルが学習されるので，最大で，276 個のモデルが利用可能である．なお，相対位置を利用する LR，WO，POS-LR，POS-WO に関しては，元の Skip-gram と異なり，文脈窓サイズ c は常に最大値 K をとるものとした．

実際の同義語判定を行う際には，学習された各 Skip-gram モデルにおいて単語間のコサイン類似度を計算する．本研究では，各モデルでの類似度を素性 (feature) とみなし，教師データに基づいて，それらの重みを線形 SVM を学習することにより，判定関数を構築する．

3 実験結果

3.1 使用データ

単語ベクトル作成において用いたコーパスとして，日本語 Wikipedia データ [1](2Gbytes) を MeCab[2] により mecab-ipadic-neologd 辞書 [3] を用いて基本形出力でわかち書きと品詞付与を行った後に，出現回数が

[1] http://dumps.wikimedia.org/jawiki/
[2] http://taku910.github.io/mecab/
[3] https://github.com/neologd/mecab-ipadic-neologd

表 2: 文脈限定 Skip-gram による同義語判定精度の評価

文脈限定関数による素性	素性の数	精度	再現率	F 値
ALL + POS	12	0.844	0.561	0.674
ALL + LR	3	0.829	0.552	0.663
ALL + WO	21	0.865	0.614	0.718
ALL + POS-LR	23	0.857	0.603	0.708
ALL + POS-WO	221	0.869	0.667	0.755
ALL + POS + POS-LR + POS-WO (MAX)	254	0.873	0.685	**0.768**
既存手法 (ALL のみ, $N = 1000$, F 値最大化)	1	0.718	0.694	0.706

100 回未満の低頻度語を除いたものを使用した．単語ベクトルが獲得された単語は 104630 種類 [4] となった．Skip-gram モデル [5] では，階層的 softmax モデルを用いて学習を行った．同義対の正例として，Wordnet 同義対データベース [6] に含まれる同義対を利用した．発生頻度が極端に低く Skip-gram で単語ベクトルの獲得できなかった単語を除き，最終的に 5848 対を正例として用いた．負例 (非同義対) としては，まず，単語ベクトルが獲得可能であった単語 [7] の中から，ランダムに作成した 17544 対 (正例の 3 倍) を利用した．更に，正例に含まれる単語群をランダムに組み合せることで作成した 5848 対 (正例と同数) を，負例として追加した．この負例の追加により，正例に含まれる特定の単語の出現のみによって同義対と誤判定してしまう問題を緩和した．

3.2 文脈限定 Skip-gram による同義語判定

ここでは，提案手法 (文脈限定 Skip-gram) による同義語判定の性能の評価実験を行った．学習時における最大文脈窓サイズ K に関しては，文脈限定の無い従来の Skip-gram モデル (ALL) については $K = 5$，他のモデルに関しては，学習対象になる周辺単語の数が減少することを考慮に入れ $K = 10$ とした．単語ベクトルの次元数 N は，すべてのモデルに関して $N = 300$ とした．2.2 節で述べたように，ある文脈限定関数について一つの素性が対応する．本研究では，表 1 の文脈限定関数の組み合わせにより素性群を作成した．なお，すべての素性群は必ず ALL を含むものとした．与えられた素性群について線形 SVM で重みを学習し，5 分割交差検定により，精度，再現率，F 値を評価した．提案手法を用いた同義語判定の結果を表 2 に示す．最初の 5 行は，ALL と一つのタイプの文脈限定関数群を組み合せた結果である．その次の行は，ALL と複数タイプの組み合わせの中で，F 値が最も高くなった結果を表示している．また，最後の行に，従来の Skip-gram モデルとの比較として，ALL のみを用いた結果を示した．この時，学習すべきパラメータは閾値のみであるため，線形 SVM ではなく，F 値最大化を用いて閾値を推定した．最大文脈窓サイズ K と単語ベクトル次元数 N についても，F 値が最大となるものを探索し，$K = 5$ および $N = 1000$ とした．従って，この F 値 (=0.706) を，従来の Skip-gram を利用して達成可能な最大の F 値とみなすことができる．表 2 において，ALL と一つのタイプのみの文脈限定関数を組み合わせた場合でも，「ALL + WO」「ALL + POS-WO」で既存手法の F 値を大きく上回ることが示されている．これは，同義語判定において，周辺単語の相対的な位置およびその分類属性が，重要な情報であるということを示唆している．また，提案手法における F 値最大となる組み合わせは，「ALL + POS + POS-LR + POS-WO」であり，F 値は 0.768 となった．これは既存手法の最大 F 値である 0.706 を大きく上回っており，提案手法の有効性を実証している．

さて，同義語判定の具体的な問題として，第 1 節において，カタカナ語と和語/漢語からなる同義対のコサイン類似度が低くなるという報告があると述べた．そこで，提案手法でこの問題が解決されるかを調べた．そこで，既存手法である ALL($N = 1000$) のみ (以下で「ALL」と参照) と，提案手法において F 値が最大であった「ALL + POS + POS-LR + POS-WO」の組み合わせ (以後,「MAX」と参照) について，カタカナ語と和語/漢語の対の同義語判定問題に関する性能を比較した．正例の同義対の中で，対の片方がカタカナ語であり，もう一方が和語/漢語のものは，2457 対

[4] 同じ単語であっても品詞が異なるものは区別して扱った

[5] https://code.google.com/p/word2vec/ にて Google が公開している実装を使用した．

[6] http://nlpwww.nict.go.jp/wn-ja/jpn/downloads.html にて NICT が提供する，Wordnet[1] を元に人手で作成された同義対データベースである．

[7] </s>は除く

表 3: カタカナ語-和語/漢語対の同義語判定における ALL と MAX の性能比較

手法	精度	再現率	F 値
ALL	0.800	0.597	0.684
MAX	0.890	0.621	0.732

表 4: MAX において判定可能となった同義対の例

一番	トップ
立案	デザイン
様式	タイプ
闘争	ファイト
脱走	エスケープ

存在した．同様に負例は 7782 対存在した．このデータセットを利用した性能比較の結果を表 3 に示す．既存手法 ALL と比べ，提案手法 MAX において，精度，再現率がともに大幅に向上している．また，具体的な成功例として，ALL では非同義対と判定され，MAX にて同義対と正しく判定された例を，表 4 に示す.「一番」と「トップ」など，既存手法において同義対と判定するのが困難だった対が，正しく判定されている．

4 結論

本研究では，同義語判定精度の向上のため，Skip-gram モデルを改良し，文脈限定関数を利用した手法を提案した．実験の結果，周辺単語の語順や品詞を考慮して文脈を限定することで，既存の Skip-gram 手法を上回る同義語判定性能が得られることを示した．また，本手法で，カタカナ語-和語/漢語の同義語判定の問題について性能が向上することを示した．本研究の成果は，辞書の単語意味データなどを利用せずに，文脈情報のみから，同義語判定の性能向上が可能であることを示したという点において，大きな意義があるものである．本手法を，既存の辞書ベースの手法 [5] や検索エンジンを利用する手法 [9] 等と組み合わせることで，さらに同義語判定精度を向上させることができると期待される．また，今後は，線形 SVM で得られた各素性に対する重みを詳細に検討し，有効な文脈限定関数を厳選することで，更なる性能向上を目指していく予定である．

参考文献

[1] Francis Bond, Timothy Baldwin, Richard Fothergill, and Kiyotaka Uchimoto. Japanese semcor: A sense-tagged corpus of japanese. In *GWC-2012*, 2012.

[2] Hagiwara Masato, Yasuhiro Ogawa, and Katsuhiko Toyama. Selection of effective contextual information for automatic synonym acquisition. In *Coling/ACL2006*, pp. 353–360, 2006.

[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS2013*, pp. 3111–3119, 2013.

[4] Harris Zellig. Distributional structure. *Word*, Vol. 10, No. 23, pp. 146–162, 1954.

[5] 笠原要, 稲子希望, 加藤恒昭. テキストデータを用いた類義語の自動作成. 人工知能学会論文誌, Vol. 18, No. 4, pp. 221–232, 2003.

[6] 乾健太郎. 自然言語処理と言い換え. 日本語学, Vol. 26, No. 13, pp. 50–59, 2007.

[7] 吉田稔, 中川裕志, 寺田昭. コーパス検索支援のための動的同義語候補抽出. 人工知能学会論文誌, Vol. 25, No. 1, pp. 122–132, 2010.

[8] 城光英彰, 松田源立, 山口和紀. 同義語判定問題を用いた語義ベクトルの評価の検討. 人工知能学会インタラクティブ情報アクセスと可視化マイニング研究会, 第 10 巻, pp. 21–25, 2015.

[9] 渡部啓吾, D. Bollegala, 松尾豊, 石塚満. 検索エンジンを用いた関連語の自動抽出. 人工知能学会全国大会論文集, 2008.

[10] 有賀竣哉, 鶴岡慶雅. 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. 言語処理学会第 21 回年次大会発表論文集 (NLP2015), pp. 752–755, 2015.