語構成情報と言い換えパターンを用いた二字漢字の句への言い換え

関沢 祐樹　　梶原 智之　　小町 守
首都大学東京
{sekizawa-yuuki, kajiwara-tomoyuki}@ed.tmu.ac.jp, komachi@tmu.ac.jp

1 はじめに

異なる表現で同じ意味を表す言い換えは、機械翻訳や情報検索など、多くの自然言語処理タスクの性能改善のために重要である。日本語の漢字は 1 文字が 1 単語を表す表語文字であり、「犯罪⇔罪を犯す」のように各漢字の意味を考慮することで単語を句へ言い換えることができる。単語から句への言い換えは読解支援、句から単語への言い換えは自動要約に直接応用もできるため有用である。日本語には、このような漢字 2 文字で構成される単語（二字漢字）が多く、例えば JUMAN 辞書（Ver.7.01）に収録されている単語約 30,000 語のうち半数の約 15,000 語が二字漢字である。本研究では、二字漢字を構成する各漢字の意味を考慮して、句への言い換えを試みる。

本研究では先行研究の手法で言い換えを生成できない二字漢字に対して、**語構成情報**を用い手動で作成した言い換えパターンを用いて言い換えを行う手法を提案する。本研究で使用する二字漢字の語構成情報は、各語構成漢字の係り受け、および語構成漢字の取りうる品詞である。本研究では、まず語構成情報を用いる言い換えパターンを手動で作成し、それを用いて二字漢字の言い換え候補を生成した。その後、分散表現を用いて二字漢字の単語ベクトルと言い換え候補の句ベクトルの類似度を計算し、二字漢字の適切な言い換えを選択した。

2 先行研究

萩行ら [1] は、国語辞典の見出し語と定義文を用いて二字漢字の言い換えを生成している。辞書定義文を JUMAN・KNP を用いて解析し、見出し語である二字漢字の語構成漢字に対応する部分（図 1 の下線部）を辞書定義文から探索後、対応部分とその間を言い換えとしている。ここで、**語構成漢字**とは、二字漢字を構成する漢字のことである。言い換え例を図 1 に示す。

萩行らは、辞書定義文から対応部分を抜き出して言い換えを生成している。そのため、この手法では二字漢字の語構成漢字と対応する表現が辞書定義文に存在しない場合、言い換えを生成できない。

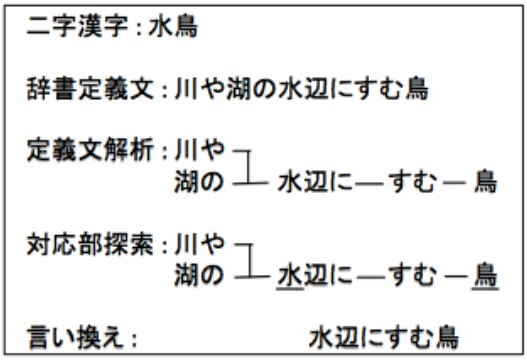

図 1: 萩行らの言い換え生成例

また、竹内ら [2] は複合名詞の係り受けを利用して複合名詞を言い換えており、竹内 [3] は TLCS（日本語の語彙概念構造）を用いてサ変名詞をガ格、ヲ格、ニ格、カラ格をもつ動詞句へと言い換えている。

本研究では言い換えパターンを用いるため、二字漢字の語構成漢字と対応する表現が辞書定義文に存在しなくても言い換えが可能である。また、言い換えパターンは、ガ格、ヲ格、ニ格、カラ格に対応するパターンを採用し、さらに、パターンを追加して言い換え、サ変名詞に限定せず二字漢字を言い換える。

3 言い換えパターンを用いた二字漢字の言い換え

我々の提案手法では、まず萩行らの手法を用いて二字漢字の言い換えを試み、言い換えを生成できなかった二字漢字に対しては**言い換えパターン**を用いた言い換えを行う。そのため、萩行らの手法よりもより多くの二字漢字に対して言い換えを生成することができる。本研究の概略を図 2 に示す。

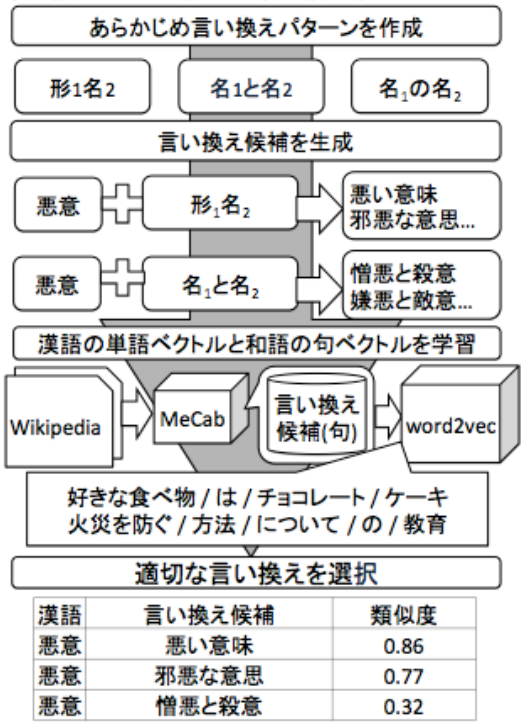

図 2: 提案手法の概略

3.1 言い換えパターンの作成

二字漢字の語構成情報を利用する言い換えパターンを作成する。言い換えパターンは、あらかじめ二字漢字の句への言い換えを考え、句の中で各語構成漢字を含む単語がどのような品詞を取るのか、どのような係り受けをするのか、どのような助詞を伴って言い換えられるのかを考慮して、人手で作成した言い換え生成ルールである。使用した言い換えパターン、および二字漢字の言い換え例を表 1 に示す。ここで名は名詞、形は形容詞、動は動詞、副は副詞、数は数量詞、形名は名詞化された形容詞であり、1 は一文字目、2 は二文字目を表している。

図 2 では、「形 $_1$ 名 $_2$」、「名 $_1$ と名 $_2$」、「名 $_1$ の名 $_2$」を例にとっている。実際は、表 1 にある全ての言い換えパターンを適用する。

3.2 言い換え候補の生成

3.1 節で作成した言い換えパターンを二字漢字に適用することで、言い換え候補を生成する。以下では二字漢字「悪意」を例として言い換える。

言い換えパターンに含まれる品詞情報を使用するため、各語構成漢字が含まれる単語を辞書から探索し、その単語と品詞を元に言い換えを生成する。単語は、JUMAN 辞書に登録されている単語全てが対象である。したがって、生成される言い換え候補も複数になることがある。言い換えパターンを「形 $_1$ 名 $_2$」とすると、形 $_1$ は「悪」を含む形容詞「悪い」、「邪悪な」「悪賢い」などに置換される。また、名 $_2$ は「意」を含む名詞「意味」、「意思」、「好意」などに置換される。したがって、生成される言い換え候補は、「悪い意味」、「邪悪な意思」、「悪賢い好意」、…となる。図 2 では、「悪意」に対して「形 $_1$ 名 $_2$」および「名 $_1$ と名 $_2$」を適用している。

[1]言い換えとして出力された数

表 1: 言い換えパターンと言い換えの例

単語	パターン	言い換え候補例	出力数 [1]
家宝	名 $_1$ の名 $_2$	家の宝	5,043
悪評	形 $_1$ 名 $_2$	悪い評判	964
縦横	名 $_1$ と名 $_2$	縦と横	495
譲位	名 $_2$ を動 $_1$	位を譲る	427
信者	動 $_1$ 名 $_2$	信じる者	220
表出	名 $_1$ に動 $_2$	表に出す	167
再会	副 $_1$ 動 $_2$	再び会う	86
脇目	名 $_1$ からの名 $_2$	脇からの目	16
一行	数 $_1$ の名 $_2$	一つの行	16
大小	形名 $_1$ と形名 $_2$	大きさと小ささ	0

生成した言い換え候補は莫大であり、中には、日本語として不自然な候補もある。例えば、「悪党と意訳」は不自然である。このような言い換え候補を除去するために、Web 日本語 N グラム第 1 版 [2] に存在しない句は全て削除した。

3.3 単語ベクトルおよび句ベクトルの構築

本研究では、二字漢字と言い換え候補がどれほど意味が似ているかを判定するために、word2vec [4] を用いて二字漢字の単語ベクトルおよび言い換え候補の句ベクトルを構築する。word2vec は、コーパスの単語の周辺文脈情報を用いて、単語をベクトルとして表すツールである。word2vec の学習のとき、MeCab[3] (Ver.0.996) を用いて学習コーパスを言い換え候補の句単位で分割することで、単語ベクトルと句ベクトルを同時に学習する [5]。

例えば「火災を防ぐ」という句が MeCab のユーザ辞書に登録されている場合、次のような分割がされる。

火災を防ぐ / 方法 / について / の / 教育

この方法では、句の文脈を考慮した句ベクトルを学習できる。ただし、コーパス中にその句が出現しない場合は、その句のベクトルは構築されない。

[2]http://www.gsk.or.jp/catalog/gsk2007-c/
[3]http://taku910.github.io/mecab/

表 2: 二字漢字 13,213 語の言い換え

	ベースライン	提案手法
言い換え生成割合	41.1%	56.0%
再現率	31.2%	43.2%

3.4 言い換え候補の選択

本研究では、単語ベクトルおよび句ベクトルの間の余弦類似度を用いて言い換えを選択する。単語ベクトルおよび句ベクトルの学習後、二字漢字の単語ベクトルと、その言い換え候補である句ベクトル全ての余弦類似度を計算する。そして、最も類似度の高い句を二字漢字の言い換えとして選択する。

4 実験

4.1 実験設定

本研究では、萩行らの辞書定義文を用いた言い換えをベースラインとする。提案手法は、ベースラインで言い換えを生成できなかった二字漢字に対して、言い換えパターンによる言い換えを行う。岩波国語辞典第五版 [4]、JUMAN 辞書 (Ver.7.01)[5] の両方に記載のある二字漢字 13,213 語を言い換えた。

また、word2vec の学習コーパスには、言い換え候補の句を全て辞書登録した MeCab の IPADic で分かち書きをした日本語 Wikipedia[6]（2015 年 11 月 3 日アクセス、5.04GB）を用いた。word2vec のパラメータは、50 次元、文脈窓 2、CBOW、階層的ソフトマックスあり、ネガティブサンプリング数 10 に設定した。MeCab は、言い換え候補の句を全て MeCab ユーザ辞書に登録した。単語コストは句の長さ × (-1800) とし、連接コストは、左は句の最左の品詞、右は句の最右の品詞とした。次に、句を登録したユーザ辞書を用いて、形態素解析器 MeCab の IPADic (Ver.0.996) によってコーパスを分かち書きする。この前処理 [4] によって、word2vec が単語ベクトルと同時に言い換え候補の句ベクトルを構築することができる。最後に、前処理を行ったコーパスに対して word2vec を適用することで、単語ベクトルと句ベクトルを学習する。

本実験では、どれくらい多くの言い換えを生成できるか、どれくらい正しい言い換えを生成できるか（再現率）によって評価を行う。正しい言い換えかどうかの評価は第一著者 1 人が行った。

[4] http://www.gsk.or.jp/catalog/gsk2010-a/
[5] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
[6] http://dumps.wikimedia.org/jawiki/latest/

表 3: 提案手法が正しく言い換える例

単語	辞書定義文	提案手法の言い換え
悪評	悪いうわさ	悪い評判
浄水	よごれのない、きれいな水	清浄な水
騒音	耳にうるさく感ずる音	騒がしい音
頭髪	あたまの毛	頭部の髪

4.2 実験結果

岩波国語辞典の定義文と、あらかじめ作成した 10 の言い換えパターンを使用して、二字漢字 13,213 語を言い換えた。生成された言い換えの数を表 2 に示す。提案手法により新たに 1,978 語に対して言い換えができるようになり、言い換え生成割合では、ベースラインと比較して提案手法が 14.9 ポイント向上した。また、二字漢字 13,213 語からランダムに選択した 500 語に対する再現率を表 2 に示す。再現率では、ベースラインと比較して提案手法が 12.0 ポイント向上した。

表 2 から適合率を見積もると、ベースラインが 75.9% に対し、提案手法は 77.1%（提案手法によってのみ生成された言い換えの適合率は 66.5%）であった。これより、ベースラインの F 値は 0.442 であり、提案手法の F 値は 0.553 であるので、ベースラインと比較して F 値が 0.111 向上した。

5 考察

5.1 提案手法

提案手法は、先行研究よりも、多くの正しい言い換えを生成した。提案手法のみが言い換えられる例を表 3 に示す。これは、先行研究が一つの辞書定義文を使用することに対し、提案手法は複数の言い換えパターンを使用したからと考える。多数の言い換え候補の中に一つでも日本語として自然なものがあれば、それが言い換えとなるからである。

5.2 言い換えパターンの出力数

表 1 は二字漢字の言い換えとして出力された句の言い換えパターンの頻度である。「名 $_1$ の名 $_2$」が最も多く出力されている。10 の言い換えパターンはあらかじめ二字漢字 150 語を手動で句へと言い換えることで作成した。言い換えパターンは徐々に追加しながら実験を行った。言い換えパターンが、表 1 の出力数上位 6 パターンのみで言い換えた場合、再現率は 41.0%であった。ベースラインと比較すると、9.8 ポイント向上した。言い換えパターンを 10 に増やすと、再現率がさらに 2.2 ポイント向上した。言い換えパターンを増やすことで、再現率の向上が期待できるが、増加量は徐々に少なくなると予測される。例えばパターン数

表 4: 英字の言い換え候補と余弦類似度

英語の頭文字	0.680
英語の字幕	0.516
英語の文字	0.410
英語の点字	0.382

を 20 に増やした場合の再現率は、さらに 2.5 ポイント向上し 45.7%になると予測される。

5.3 正しい言い換えを生成できない例

提案手法で正しい言い換えを生成できない二字漢字の中には、海豚のような当て字などが存在した。このような単語は、ランダムに選択した二字漢字 500 語のうち 27 単語存在し、言い換えが存在しないものである。

5.4 二字漢字と句の文脈の分布

学習コーパスにおいて、二字漢字とその言い換えの周辺文脈が異なっている場合がある。「英字」という二字漢字を例に挙げる。予想される言い換えは「英語の文字」である。Web 日本語 N グラム第 1 版において、「英字」の前には「半角」が多く、「英字」の後には「で　入力」、「新聞」が多い。一方「英語の文字」の前には「楽しい」、「日本語　や」が多く、「英語の文字」の後には「学習」が多い。このように周辺文脈が異なっているため、二字漢字と似た意味の句のベクトルが、二字漢字の単語ベクトルに類似するベクトルとして学習されていない可能性がある。表 4 は「英字」の言い換え候補と余弦類似度である。本研究では、JUMAN 辞書を用いて言い換え候補を生成しているため、「頭文字」が 1 つの単語として使用される。そのため、「英語の頭文字」が言い換え候補として生成される。「頭」と「文字」の 2 単語として処理できれば、「頭文字」が 1 単語として処理されなくなるため、「英語の頭文字」が言い換え候補からなくなると考える。

5.5 言い換えを選択できない例

提案手法は句ベクトルの学習を行うため、句の文脈を見ることができる。しかし、Wikipedia の中には言い換えパターンから生成した表現にマッチする句が少なく、句ベクトルをうまく学習できなかった。図 3 は、ランダムに選択した二字漢字、言い換え候補の句 500 語の Wikipedia での頻度である。単語の頻度よりも句の頻度は低く、ほとんど出現しない句が多い。

また、4 節で正しい言い換えかどうかを判定する為に使用した二字漢字 500 語のうち 38 単語は、正しい言い換えが生成されている一方、その言い換えの句ベクトルが学習されておらず、言い換えとして選択されなかった。これは、ベクトルの学習の際、言い換え候補の句が出現しなかったためである。

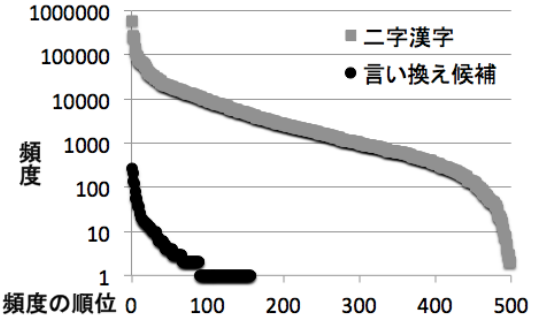

図 3: Wikipedia 中の二字漢字と句の頻度

6　おわりに

本研究では、二字漢字を構成する各漢字の意味を考慮して、句への言い換えを行った。提案手法は先行研究の手法で言い換えを生成できない二字漢字に対して、手動で作成した言い換えパターンを用いて言い換えを行う手法である。提案手法は先行研究よりも 14.9 ポイント多くの言い換えを生成し、12.0 ポイント多くの正しい言い換えを生成し、F 値が 0.111 向上した。

これからの課題は、言い換えパターンを自動的に獲得することである。本研究では人手で作成したため、全ての二字漢字を言い換えることができなかったが、自動で獲得することでより多くの二字漢字を言い換えできると考える。また本研究の手法では、二字漢字のみを言い換えている。今後は文字数に縛られずに単語を句へと言い換えたい。

参考文献

[1] 萩行正嗣, 黒橋禎夫, 辞書定義文を用いた二字漢語の言い換え表現の生成, 言語処理学会第 15 回年次大会発表論文集, pp.256-259, 2009.

[2] 竹内孔一, 内山清子, 吉岡真治, 影浦峡, 小山照夫, 語彙概念構造を利用した複合名詞内の係り関係の解析, 情報処理学会論文誌, Vol.43, No.5, pp.1446-1456, 2002.

[3] 竹内孔一, 語彙概念構造による動詞辞書の作成, 言語処理学会第 10 回年次大会発表論文集, pp,576-579, 2004.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In ICLR, 2013.

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In NIPS. pp.3111-3119. 2013.

[6] Quoc V Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. In ICML. pp. 1188-1196. 2014.