

文脈を考慮した句の平易化

河原井 翼
北陸先端科学技術大学院大学
s1910074@jaist.ac.jp

白井 清昭
北陸先端科学技術大学院大学
kshirai@jaist.ac.jp

1 はじめに

テキストの平易化は、文もしくは文章を同じ意味を持つ易しい表現に言い換える技術である。難解なテキストを子供や日本語を母語としない人にも理解しやすいようなテキストに直すなど、多くの応用場面がある。2節で論じるように、日本語を対象とした平易化の先行研究の多くは単語の言い換えを対象としていた。すなわち、文中に出現する難しい単語を同じ意味を持つ易しい単語に置き換える。しかしながら、人が文を易しく言い換える際には、句、節、あるいは文全体を言い換えることもあり、単語単位の言い換えだけでは不十分である。本研究では、句を単位にテキストを平易化する手法を探究する [1]。ただし、平易化の対象とする句は、連続する「名詞-助詞-動詞」という単語列に限定する。句を単位とした言い換えにより、多様な平易化テキストを生成することを狙う。さらに、文脈を考慮して難解な句を平易な句に適切に言い換える手法を探究する。

2 関連研究

梶原と山本は、国語辞典の語釈文から平易な単語の言い換えルールを獲得することで、小学生のための文書読解支援に向けたテキストの平易化を実現している [2]。言い換え後の平易な単語を選択するための様々な手法を比較し、シソーラスに基づく単語の類似度を用いた平易化が最も良いと結論づけている。Krizらは、語彙の特徴と文脈の特徴を利用した難解語抽出モデルによって難解語を検出し、文脈を考慮した平易な単語への言い換えモデルを実現している [3]。文脈を考慮することで従来手法よりも高い精度で難解語を抽出でき、また文脈に応じた好ましい言い換えが実現できることを示した。梶原と小町は、難解なテキストと平易なテキストからなるパラレルコーパスを自動構築し、これを訓練データとして難解なテキストを平易なテキストに翻訳するモデルを学習することで平易化を実現している [4]。

先行研究 [2, 3] は、単語単位の言い換えを実現する手法であり、句を対象とした平易化ではない。先行研究 [4] は文全体を言い換えるモデルであり、単語だけでなく句の言い換えも含まれるが、ユーザの要求に応じて平易化する句としない句を分けるといった調整が難しい。これらに対し、本研究では、単語や文ではなく難解な句を入力として、これを平易な句に言い換える手法を探究する。

3 提案手法

3.1 概要

ある文脈に「 N - P - V 」という句が出現するとき、これを平易かつ同じ意味を持つ別の句に言い換える。ここで N は名詞、 P は助詞、 V は動詞である。平易化の際には文脈を考慮する。すなわち、句 N - P - V を言い換えたとき、それを含む元の文の意味が変わったり不自然にならないことに留意する。

言い換えは以下の3種類の手法で実現する。

- 名詞のみの言い換え
名詞 N をより平易な別の名詞に言い換える。
(例) 果敢-に-挑む → 積極的-に-挑む
- 動詞のみの言い換え
動詞 V をより平易な別の動詞に言い換える。
(例) 果敢-に-挑む → 果敢-に-挑戦する
- 名詞、動詞の言い換え
 N 、 V ともに平易な別の単語に言い換える。
(例) 果敢-に-挑む → 積極的-に-挑戦する

これらの手法で生成した言い換え候補のそれぞれについて、その妥当性をスコア付けし、最終的に最適なものをひとつ選択する。

3.2 平易言い換え単語対の収集

名詞もしくは動詞を平易に言い換えるため、難解語と平易語の組から構成される平易言い換え単語対のデータベースを用意する。データベースは以下の

2つの辞書を用いて構築する。

- SNOW D2: 内容語換言辞書 [5]

形態素解析ツール Juman の内容語辞書項目を人手で言い換えた辞書。この辞書では、言い換え後の単語は必ずしも言い換え前よりも平易ではない。そこで、日本語教育語彙表 [6] を参照し、言い換え前、後の語が日本語教育語彙表に掲載されていて、かつ言い換え後の単語のレベルが言い換え前の単語のレベルよりも低い組を選別する。言い換え対の数は 1,734 である。

- Simple PPDB : Japanese [7]

日本語単語の平易な言い換え辞書。難解語、平易語、難解語・平易語の難易度、言い換え確率の組から構成される。本研究では、言い換え確率が 0.1 以上のデータを選別して使用する。また、前述の SNOW D2 にあわせて、難解語、平易語がともに日本語教育語彙表に掲載されていて、かつ言い換え後のレベルが低い組を選別する。言い換え対の数は 10,351 である。

3.3 平易化する句の検出

与えられた文の形態素解析を行う。形態素解析ツールとして MeCab[8]、辞書として mecab-ipadic-NEologd[9] を用いる。次に、名詞、助詞、動詞の連続を句として抽出する。この際、動詞は基本形に変換する。名詞、動詞がともに平易言い換え単語データベースの難解語として登録されているとき、平易化の対象句とする。

3.4 平易句のスコア付け

平易化の対象句 $Ph = N-P-V$ に対し、平易言い換え単語データベースを用いて、名詞のみ、動詞のみ、名詞と動詞の両方を別の単語に言い換えて、言い換え後の平易句の候補 $Ph'_i = N'_i-P-V'_i$ を得る。 Ph'_i のそれぞれのスコアを計算し、それが最大となる Ph'_i を選択する。スコアは2つの観点から算出する。ひとつは、正確性、すなわち言い換え前後で意味を保持しているかどうかという観点、もうひとつは流暢性、すなわち言い換え後の句がどれだけ自然な句であるかという観点である。

3.4.1 正確性のスコア

S を元の句 Ph を含む文、 S'_i を Ph を Ph'_i に言い換えた後の文とする。正確性 (faithfulness) のスコア

$FA(Ph'_i)$ を式 (1) のように定義する。

$$FA(Ph'_i) = \cos(\vec{S}, \vec{S}'_i) \quad (1)$$

\vec{S}, \vec{S}'_i は文 S, S'_i の分散表現、 \cos はコサイン類似度である。すなわち、平易化前の文と後の文の分散表現が似ているほど、 Ph'_i の正確性は高いとみなす。言い換え対象の句だけでなくそれを含む文全体の類似度を考慮することで、言い換え後の平易句が文脈に沿って適しているかを評価する。

文の分散表現は Sentence BERT[10] を用いて得る。Sentence BERT は、事前学習した BERT(Bidirectional Encoder Representations from Transformers)[11] を基に、Siamese Network を用いた fine-tuning によって文の分散表現を学習するモデルである。本研究では Sentence BERT 日本語モデル [12] を利用する。

3.4.2 流暢性のスコア

言い換え後の句 Ph'_i の流暢性 (fluency) のスコア $FL(Ph')$ を式 (2) のように定義する。

$$FL(Ph'_i) = \frac{\text{count}(Ph'_i)}{\sum_j \text{count}(Ph_j)} \quad (2)$$

式 (2) の分子は句 Ph'_i のコーパスにおける出現頻度、分母は全ての句の出現頻度の和である。すなわち、句 Ph'_i がコーパスによく出現するほど、その句は自然な句であるとみなす。

句の出現頻度 $\text{count}(Ph_j)$ は京都大学格フレーム Ver 2.0 を用いて算出する。このデータはウェブから収集した日本語 100 億文から自動構築した格フレームであり、用言と格要素の頻度、すなわち $N-P-V$ という句の頻度の情報も付与されている。本研究では $N-P-V$ の頻度をそのまま $\text{count}(Ph_j)$ として用いる。

3.4.3 平易句のスコアの定義

最終的に句のスコアは $FA(Ph')$ と $FL(Ph')$ の対数の重み付き和で算出する。この際、式 (3) と (4) の2通りのスコアを提案する。

$$\text{Score1}(Ph'_i) = \alpha \log FA(Ph'_i) + \log FL(Ph'_i) \quad (3)$$

正確性のスコア $FA(Ph'_i)$ は言い換え前後の文の分散表現のコサイン類似度なので、多くは 1 に近い値を取るのに対し、流暢性のスコア $FL(Ph'_i)$ は句の相対出現頻度であるため、 1.0×10^{-8} 付近の小さい値を取る。Score1 では、パラメタ α は両者のスケールを合わせるために用いる。正確性と流暢性のスコアを同程度に評価することに注意していただきたい。

α は次のように決める。4.1 項で後述する開発データ (元の句とそれを平易化した句の集合) における全ての事例について $FA(Ph')$, $FL(Ph')$ を算出し, それぞれの中央値 Me_{fa} と Me_{fl} を求め, $\alpha = \frac{Me_{fl}}{Me_{fa}}$ とする。

$$Score2(Ph'_i) = \beta \alpha \log FA(Ph'_i) + (1 - \beta) \log FL(Ph'_i) \quad (4)$$

$Score2$ におけるパラメタ α は, $Score1$ と同様に, 2 つのスコアのスケールを合わせるためのものであり, 前述のように開発データを用いて決める。一方, パラメタ β は, 正確性と流暢性のスコアに対する重み付けである。

3.5 句の選択

予備実験では, 式 (3) または (4) が最大となる句を選択したところ, 名詞と動詞の両方を言い換える句よりも, 名詞もしくは動詞のいずれかを言い換えた句が多く選ばれる傾向が見られた。これは, 正確性のスコア FA は言い換え前後の文の類似度なので, 単語を 2 つ置換した場合よりも 1 つだけ置換した場合の方が高くなる傾向があるためと考えられる。一方, 本研究の目的は句を対象とした言い換えであり, 名詞と動詞の両方を言い換えた句をできるだけ生成したい。そこで, 以下の方式により言い換え候補の中から句を選択する。

1. 名詞と動詞の両方を言い換えた平易句のスコアの値が閾値 T よりも大きいとき, つまり平易化された句として妥当である可能性が高いとき, それを選択する。
2. そのような句がないとき, 全ての候補の中からスコアが最大のものを選択する。

閾値 T は $Score1$ と $Score2$ とで個別に設定する。設定方法については 4.2 項で述べる。

4 評価実験

4.1 データセット

まず, 開発データを作成した。毎日新聞の 2013 年の記事から, 3.3 項で述べた手法を用いて平易化の対象とする句, およびそれを含む文を 500 件抽出した。これらの句を提案手法によって平易な句に言い換えた。さらに, 正確性を満たすか否か, 自然さを満たすか否か, (両者を考慮して) 平易句として妥当であるか否か, を人手で判定した。この開発デー

タは, パラメータ α や T の最適化に用いた。

次に, 評価データを作成した。毎日新聞の 2013 年の記事から平易化の対象とする句を 100 件抽出した。これらは開発データとは異なる句を選んだ。評価データに対し, 著者 2 名が独立して以下のアノテーションを行った。

- 元の句 Ph を含む文, およびそれを平易句の候補 Ph'_i に置き換えた文を全て提示し, その中から平易な言い換えとして最も適切なものを選んだ。ここで最も適切な平易句とは, 子供や日本語を非母語話者とする人に最もわかりやすい表現とした。
- 提案手法によって生成された平易句¹⁾に対し, 平易さ (元の句と比べて易しくなっているか), 意味の保持 (元の句と比べて意味が変わっていないか), 自然さ (言い換え後の文が自然か) という 3 つの観点から, 1~5 までの評点をつけた。

4.2 パラメータの決定

3.4.3 で述べたパラメータ α の決定について述べる。開発データに対して生成された全ての平易句の候補についての $\log FA(ph'_i)$ と $\log FL(ph'_i)$ の中央値は, $Me_{fa} = -0.0111$, $Me_{fl} = -16.9$ となった。これに従い, $\alpha = \frac{Me_{fl}}{Me_{fa}} = \frac{-16.9}{-0.0111} = 1.54 \times 10^3$ と決めた。

3.5 項で述べた, 名詞, 動詞の両方を言い換えた平易句を優先して選択するための閾値 T の決定について述べる。基本的に, 閾値 T は, 生成した平易句が適切である可能性が高いスコアの値を設定する。 $Score1$ を用いるとき, 開発データにおいて, 正確性のスコアが閾値 t のときに平易句が正確性を満たすと判定したとき, t を X 軸, 判定精度を Y 軸としたグラフを得る (図 1(a))。同様に, 流暢性について閾値 t を変化させたときの判定精度の変化も調べる (図 1(b))。これらの精度は, 開発データのうち名詞, 動詞を両方言い換えた句が選択されたデータのみを使用して算出する。これら 2 つのグラフでは, 閾値 t を大きくすると精度が高くなる傾向が見られる。精度が十分に大きいとき, 具体的には図 1(a) で精度が 0.7 になるときの閾値 T_{fa} は 0.994, 図 1(b) で精度が 0.8 になるときの閾値 T_{fl} は 2.33×10^{-7} であった。これらを式 (3) に当てはめ, $T = \alpha \times \log T_{fa} + \log T_{fl} = -24.5$ と設定した。

同様に, $Score2$ を用いて句を選択する際の閾値

1) スコアが最大の句のみ

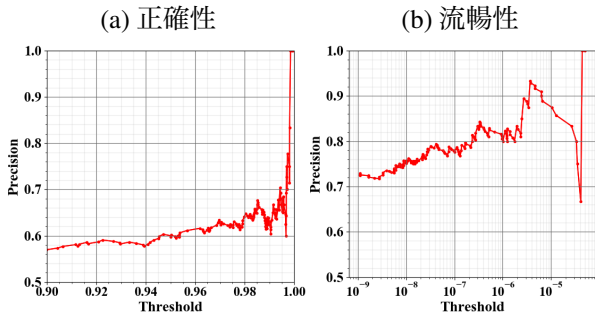


図1 開発データにおける平易句判定の精度 (Score1)

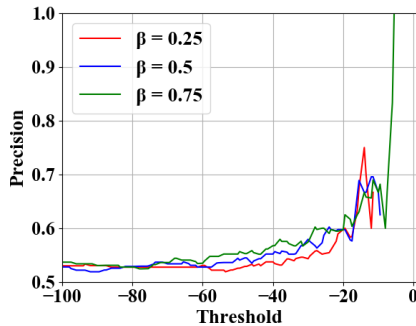


図2 開発データにおける平易句判定の精度 (Score2)

T を決定する。開発データにおいて、式 (4) に示す $Score2$ の値が閾値 t を越えたときに平易句が妥当であると判定したとき、 t を X 軸、判定精度を Y 軸としたグラフを図 2 に示す。 $\beta = 0.25, 0.5, 0.75$ のそれぞれについて、精度が 0.65 に達したときの閾値 $-15.5, -16.0, -14.0$ を T と設定した。

4.3 実験結果と考察

提案手法では、複数の平易句の候補の中からスコア付けに基づいて平易句をひとつ選択するが、この性能を評価する。具体的には、システムが選択した平易句が人手によって選択された正解とどれだけ一致しているかを正解率と定義し、これを測る。表 1 は、スコア付けとして正確性のスコア $FA(Ph'_i)$ のみ、流暢性のスコア $FL(Ph'_i)$ のみ、両方 ($Score1$ または $Score2$) を用いたシステムについて、2 名の評価者を正解としたときの正解率を示している。

表1 平易句の選択手法の評価 (正解率)

システム	評価者 1	評価者 2
$FA(Ph'_i)$	0.45	0.43
$FL(Ph'_i)$	0.34	0.38
$Score1$	0.41	0.44
$Score2 (\beta = 0.25)$	0.47	0.44
$Score2 (\beta = 0.5)^{2)}$	0.41	0.48
$Score2 (\beta = 0.75)$	0.44	0.51

正確性あるいは流暢性のスコアのみを使用したときよりも、両方を考慮したスコア ($Score1$ や $Score2$) の方が正解率が高い傾向が見られる。特に評価者 2 についてその傾向が顕著である。 $Score1$ と $Score2$ を比較すると、 $Score2$ の正解率は $Score1$ よりも同等もしくはそれ以上であることから、 $Score2$ の方が適切な平易句を選択するのに適していると言える。 $Score2$ における異なる β の比較については、評価者 1 と評価者 2 で優劣の傾向が異なるため、どちらが最適を決めることはできない。 β の最適化は今後の課題である。

この実験では複数の平易句の中から最良のものを選択しているが、スコアが最大の平易句ではなくても易しい言い換えとして妥当な句が生成されていることも多く、何が最良の平易句なのかは判断が難しい。評価者 1 と評価者 2 で同じ平易句を選択した割合は 0.61 であり、決して高くはない。提案システムの最高の正解率は 0.51 であり、二者の判定の一致率の 84% に達している。

提案手法によって生成された平易句を「平易さ」「意味の保持」「自然さ」の観点から評価した結果を表 2 に示す。どの観点についても、おおむね評点は 4 点以上であり、ある程度品質の高い平易句が生成されていることを確認した。また、表 1 の正解率が全般的に低いのにに対し、表 2 の評点が比較的高いのは、既に述べたように、システムが選択した平易句が人手で選ばれた平易句と一致していなくても妥当であるとみなせる場合が多かったためである。

表2 平易句の品質評価

	平易さ	意味の保持	自然さ
評価者 1	4.06	4.02	4.29
評価者 2	3.79	4.38	4.29
平均	3.93	4.20	4.29

5 おわりに

本論文は、名詞-助詞-動詞という句を平易に言い換える手法を提案し、その有効性を実験により示した。今後は、正確性や流暢性を測る指標を洗練し、より適切に平易句を生成する手法を探究したい。特に、現時点での流暢性は句の出現頻度だけを指標としており、平易化した句に置き換えた文全体がどれだけ自然かを評価するように改善するべきである。

2) 閾値 T の決め方が異なるため、 $Score1$ と $Score2(\beta = 0.5)$ は同じシステムではない。

参考文献

- [1]河原井翼. 句の言い換えによるテキストの平易化. 修士論文, 北陸先端科学技術大学院大学, 3 2021.
- [2]梶原智之, 山本和英. 語釈文を用いた小学生のための語彙平易化. 情報処理学会論文誌, Vol. 56, No. 3, pp. 983–992, 2015.
- [3]Reno Kriz, Eleni Miltsakaki, Marianna Apidianaki, and Chris Callison-Burch. Simplification using paraphrases and context-based lexical substitution. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 207–217, 2018.
- [4]梶原智之, 小町守. 平易なコーパスを用いないテキスト平易化. 自然言語処理, Vol. 25, No. 2, pp. 223–249, 2018.
- [5]山本和英, 吉倉孝太郎. 用言等換言辞書を人手で作りました. 言語処理学会第 19 回年次大会, pp. 276–279, 2013.
- [6]日本語教育表 ver 1.0. <http://jhlee.sakura.ne.jp/JEV.html>. (2020 年 12 月閲覧).
- [7]梶原智之, 小町守. Simple PPDB: Japanese. 言語処理学会第 23 回年次大会, pp. 529–532, 2017.
- [8]MeCab: Yet another part-of-speech and morphological analyzer. <http://taku910.github.io/mecab/>.
- [9]佐藤敏紀, 橋本泰一, 奥村学. 単語分かち書き辞書 mecab-ipadic-NEologd の実装と情報検索における効果的な使用方法の検討. 言語処理学会第 23 回年次大会, pp. 875–878, 2017.
- [10]Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3973–3983, 2019.
- [11]Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.
- [12]園部勲. Sentence BERT 日本語モデル. [https://](https://qiita.com/sonoisa/items/1df94d0a98cd4f209051)

qiita.com/sonoisa/items/1df94d0a98cd4f209051.
(2020 年 12 月閲覧).