情報処理学会論文誌 Vol.62 No.9 1605–1619 (Sep. 2021)

説明文を対象とした日本語文末述語の平易化

加藤 汰一[1,a)] 宮田 玲[1,b)] 佐藤 理史[1]

受付日 2021年1月6日, 採録日 2021年6月7日

概要：日本語文の文末述語は，内容語とアスペクト・モダリティ・丁寧体などの機能表現の複雑な組合せからなることが多く，それがしばしば日本語学習者によるテキスト読解を妨げる要因となる．従来の語彙平易化手法の多くは，難解な語を単語単位で平易な同義語に置き換える枠組みを採用しており，文末述語の平易化には必ずしも適していない．そこで本研究では，難解表現の検出および換言候補の生成・検証・ランキングからなる基本的な語彙平易化のプロセスを採用しつつ，日本語文末述語を一括して平易に言い換える手法を提案する．本手法の最大の特徴は，換言候補の生成プロセスにおいて事前学習済みのマスク言語モデルである BERT を効果的に適用することで，文全体の主要な意味を保持したまま，文末述語をまとめて平易化することである．これにより多様な表現候補の生成が可能となる．説明文を対象とした人手評価実験の結果，提案手法は複数の従来手法と比較して，一貫して多くの流暢かつ妥当な換言候補を生成できることが示された．さらに，(1) 平均トークン埋め込みとドロップアウトの有効性，(2) 生成された候補の平易度，(3) 適用先テキストドメインによる性能の違い，(4) 提案手法のエラー事例を詳細に調査することで，提案手法の挙動の特徴や改善点を明らかにした．

キーワード：語彙平易化，マスク言語モデル，言い換え生成，人手評価，エラー分析

Simplification of Japanese Sentence-ending Predicates in Descriptive Text

Taichi Kato[1,a)] Rei Miyata[1,b)] Satoshi Sato[1]

Received: January 6, 2021, Accepted: June 7, 2021

Abstract: Japanese sentence-ending predicates tend to be composed of a complex sequence of content words and functional elements, such as aspect, modality, and honorifics, which can often hinder the understanding of language learners. Conventional lexical simplification methods, which are designed to replace difficult target words with simpler synonyms in a word-by-word manner, are not always suitable for simplifying such Japanese predicates. Here, we propose a novel method that can simplify the whole sequence of predicate, following a basic lexical simplification process consisting of detection, generation, validation and ranking steps. The principal feature of our method is the high ability to substitute the whole predicates with simple ones while maintaining their core meanings in the context by effectively using the pre-trained masked language model of BERT. Experimental results showed that our proposed method consistently produced many more candidates that are both fluent and adequate than the multiple baseline methods. Furthermore, we conducted in-depth analyses of (1) the effectiveness of the average token embedding and dropout, (2) the simplicity of generated candidates, (3) the differences of performance by text domain, and (4) the remaining errors of our proposed method, revealing the characteristics of our methods and future prospects for improvement.

Keywords: lexical simplification, masked language model, paraphrase generation, human evaluation, error analysis

1. はじめに

語彙平易化とは，テキストをより理解しやすくするために，意味を保持したまま難解語を平易な語や句に置き換える

[1] 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University, Nagoya, Aichi 464–8603, Japan
[a)] kato.taichi@j.mbox.nagoya-u.ac.jp
[b)] miyata.rei@c.mbox.nagoya-u.ac.jp

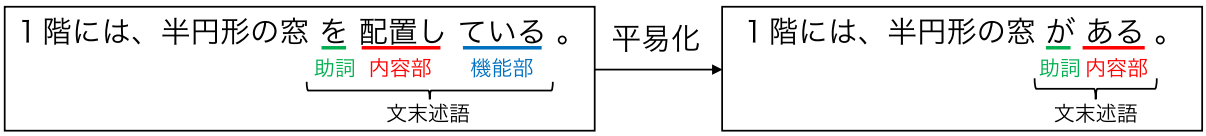

図 1 日本語文末述語とその平易化の例
Fig. 1 Example of Japanese sentence-ending predicate and its simplified version.

タスクである [1], [2]．語彙平易化は，子どもや第二言語学習者のテキスト読解を促進する用途だけでなく [3], [4], [5]，機械翻訳などの自然言語処理タスクの前処理にも有効であると報告されている [6]．一般的な語彙平易化手法は，様々な言語資源から構築した，難解語と平易語の対からなる言い換え語リストを使用する．近年は，word2vec [7] や BERT [8] などの汎用的な事前学習モデルを活用して，言い換え語の候補を獲得する試みも増えている [9], [10], [11]．これらの従来手法は様々な文に広く適用することができるものの，あくまで単語（ないし句）ごとの局所的な言い換えを順次適用するものであり，複数の言語要素をまとめて柔軟に言い換えることができない．

本研究で扱う日本語文末述語は，**図 1** に示すように，助詞，内容部（主に動詞，名詞，形容詞），機能部（主に助詞，助動詞）の 3 つの要素からなるものとする[*1]．機能部は，アスペクト・モダリティ・丁寧体などが複雑に結合しており，表現を難解にする要因となる．図 1 は，述語を人手で平易化した例も示している．この例では述語全体が，先頭の助詞も含めて，包括的に書き換えられている．言い換え語リストを使用した語単位の手法では，「配置する」に対して「設ける」，「取りつける」といった類義語を出すことはできるが，この例のような平易化を再現することは原理的に難しい．文末述語の平易化においては，複数単語を包括的かつ柔軟に言い換える仕組みが求められる．

図 1 で注目すべきは，原文の内容語「配置する」が，必ずしも語単位での同義語とはいえない「ある」に変化し，情報が一部落ちているにもかかわらず，文の主要な意味は保持されている点である．我々は，このような柔軟な言い換え方法は，対象物の存在や状態を表現する**説明文**[13] において広く適用可能と考える．説明文では，主要な情報は述語（主に動詞）より，項（主に名詞）とその修飾語（主に形容詞）に集約されている．事実，説明文から述語が省略されても，人間は残りの部分だけを参照することで，文の意味を保持したまま，ある程度自然かつ妥当な代替表現を補完することができる．以上をふまえ，本研究では，文中のマスクされたトークンを予測できるマスク言語モデルである BERT を活用して，人間の言語的直感を再現しながら，柔軟な換言候補を生成する手法を提案する．具体的には，全体がマスクされた述語部分を BERT により順次補完した後に，機能表現の形を整えることで，候補を生成する．その際，適度に関連性のある幅広い言い換え候補を獲得するために，平均トークン埋め込みとドロップアウトを用いて元の述語の意味を部分的に符号化する．また，BERT により獲得する候補は必ずしも平易な表現のみとは限らないため[*2]，複数の候補を取得したうえで難解な候補を除外し，さらに流暢性・妥当性を考慮した候補のランキングを行う．

以下，2 章では，テキスト平易化，語彙平易化に関する先行研究を紹介し，日本語文末述語に適用する際の問題点を説明する．3 章では，文末述語の平易化プロセスの全体フローを定義した後に，BERT を活用した日本語文末述語の平易化手法を提案する．4 章では，実験設定として提案手法とベースライン手法の実装の詳細および評価手法を説明する．5 章では，平易化プロセスの各ステップで得られた候補の統計と評価結果を示し，提案手法の高い有効性を示す．6 章では，評価結果を基に詳細な分析を行い，提案手法の適用可能性や限界を明らかにしながら，手法の改良の方向性を提示する．7 章では，本論文の成果をまとめ，今後の研究開発や実用化の展望を述べる．

なお本論文は，国際会議 The 13th International Conference on Natural Language Generation で発表した論文 [15] の全体の論旨を見直し，手法や結果の説明を拡充したうえで，ベースライン手法の追加（4.1.2 項），人手参照文との比較評価（5.2.2 項），生成された候補の平易度調査（6.2 節），適用先テキストドメイン間の性能比較（6.3 節）を新たに加えたものである．

2. 関連研究

テキスト平易化は，意味を保ったまま複雑なテキストを

[*1] なお述語の定義と範囲に関する見解は研究者によっても揺れているが，本研究では Kawahara ら [12] による日本語述語の定義を拡張し，内容部に先行する助詞まで含めた．

[*2] 本研究では，一定数の候補を取得すれば，平易な候補は含まれるという想定の下で実験を行う．なお，Wikipedia など一般向けのテキストで学習した BERT がもっともらしいものとして予測する語は，「一般によく使われる語」になりやすいことが予想される．しかし，それはあくまで仮説であり，具体的な検証が必要である．さらに，「一般によく使われる語」が平易な語であるかについても検証の余地がある．たとえば，旧日本語能力検定の出題基準 [14] に基づく語彙リスト中の 5,136 語を対象に，平易度（2〜4 級）と日本語版 Wikipedia（https://dumps.wikimedia.org/ より取得した 2019 年 10 月版を使用）における頻度の相関を予備的に調査したところ，ピアソンの積率相関係数は 0.040，スピアマンの順位相関係数は 0.119 と低い値になった．本論文の射程外ではあるが，BERT などの事前学習モデルが，どのような条件においてどの程度平易な候補を出せるかの調査は，今後の重要な課題である．

より平易なテキストへ変換するタスクである．近年のテキスト平易化研究は，単言語の機械翻訳タスクとして定式化される場合が多く [16], [17], [18]，強化学習を活用した手法も提案されている [19]．Seq2seq ニューラル機械翻訳は平易化性能の向上に貢献したが，原文とそれに対応した平易文をセットにした大規模なパラレルデータが必要である．英語では，English Wikipedia と Simple English Wikipedia[*3]から自動で文をアライメントすることによって構築された PWKP データセット [20] や，人手でニュース記事を言い換えることによって構築された Newsela コーパスなどが存在する [21]．

日本語でも，クラウドソーシングを利用して平易化のためのデータセットを構築する試みがいくつか行われている [22], [23]．しかし，入手可能な英語の言語資源と比べると，日本語のものは頑健なモデルを構築するためには十分であるとはいえない．テキスト平易化のためのアライメントされたコーパスに乏しい言語では，機械翻訳ベースの手法の有効性は限定的である．

語彙平易化は難解語を平易な語や句に置き換えるテキスト平易化のサブタスクである．最も一般的な語彙平易化手法は WordNet や他の資源から抽出した平易な同義語を活用するものである [24]．Simple PPDB [25] は，PPDB [26] から構築された難解語と平易な同義語がペアになった大規模なデータセットである．日本語でも，同様の研究がいくつか報告されている [3], [27]．対象となる語の同義語を得るために，単語埋め込みを利用した研究もある [9], [28]．しかしながら，これらの語彙平易化手法はいずれも語単位の平易化手法であり，日本語文末述語のようなより長いテキストスパンをまとめて柔軟に言い換えるタスクには必ずしも適さない．

Seq2seq の変換手法によらず，語より長いテキストスパンを扱うためには，大規模データで学習された汎用の単言語モデルを活用することが有効だと考える．近年，様々な種類の事前学習モデルが提案され，多くの言語で訓練されてきた [29]．日本語でも，Transformer ベースのアーキテクチャで，マスク言語モデリング（masked language modeling）と隣接文予測（next sentence prediction）という 2 つのタスクにより学習させた BERT（Bidirectional Encoder Representations from Transformers）[8] など，各種の言語モデルが手に入る．1 章で述べたように，とりわけ，文中でマスクされた単語を推測する能力を持つ BERT のようなマスク言語モデルは，平易化タスクにも有効に使えることが期待できる．近年は，対象語と意味的にも文脈的にも適したより平易な語を生成する BERT ベースの語彙平易化手法も提案されているが [10], [11]，従来の語彙平易化研究と同様，平易化の単位が語レベルである．すなわち，より長いスパンのテキストを平易化する際のマスク言語モデルの適用可能性については，まだ調査されていない．またパラレルコーパスなどの教師データを利用しない事前学習モデルのみで，どのようなテキストを対象にどの程度まで平易化タスクが解けるか，あるいは解けないかに関する詳細な分析も進んでいない．本研究は，説明文というテキストタイプを対象に，文末述語という比較的長く複雑な言語要素の平易化タスクにおけるマスク言語モデルの有効性と限界を，様々な角度から検証する試みであり，今後の研究の試金石となりうるものである．

3. 手法

本研究における平易な文末述語とは，次に示す 2 つの基準を満たすものとし，逆に基準を満たさないものは難解な文末述語と呼ぶ．

- 内容部の動詞・名詞・形容詞・副詞が旧日本語能力試験（旧 JLPT）2 級までの語彙に含まれる[*4]．
- 機能部の機能語列が旧 JLPT 3 級までの文法事項に含まれる[*5]．

言い換えの適否の基準は，(1) 文法的な誤りが生じていないかと，(2) 文全体で同義性が保たれているかである．先行研究 [6] における人手による平易化タスクでも，図 1 に示したような柔軟な書き換えが試みられており，変換対象の文末述語単体の局所的な同義性にとらわれすぎないことが肝心である．以上をまとめると，本研究の平易化タスクとは，入出力の単位を文とし，上記の難易度の基準を満たさない難解な文末述語を，文全体の主要な意味が保たれる範囲で，文法的に正しい平易な文末述語に言い換えることである．

なお，説明的な文章においては疑問文や命令文も出現しうるが，本研究における説明文は平叙文（体言止めも含む）で書かれたもののみを対象とする．

3.1 平易化プロセスの全体像

一般的な語彙平易化プロセス [30] に従い，以下の 4 つのステップからなる平易化プロセスを提案する．

(1) 難解表現の検出
(2) 言い換え候補の生成
(3) 言い換え候補の検証
(4) 言い換え候補のランキング

図 2 にプロセスの全体像を示す．なお，検出ステップの前に簡易な前処理工程を設け，入力文の 3 つのタイプの機

[*3] https://simple.wikipedia.org/

[*4] 2010 年に開始された新試験（https://www.jlpt.jp/）では，語彙や文法項目のリストを掲載した出題基準が公開されていない．そこで，旧試験向けの出題基準 [14] における語彙リストを使用した．なお本研究で使用した語彙リストには，4 級の語が 740 語，3 級の語が 688 語，2 級の語が 3,708 語含まれる．

[*5] 同じく旧試験向けの出題基準 [14] に基づき，文末の機能表現に限定し，4 級のパターンを 50 種類，3 級のパターンを 35 種類定義した．

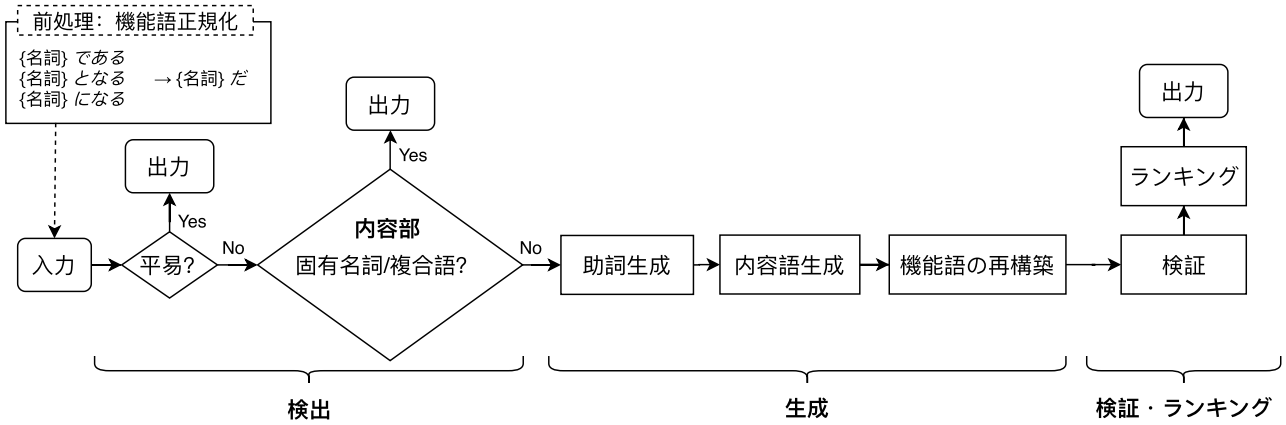

図 2 文末述語平易化のプロセス

Fig. 2 Overall process of simplification of sentence-ending predicates.

能表現（「である」,「となる」,「になる」）を正規化する[*6]．以下では，「1 階には、半円形の窓を配置している。」という 1 文を入力例として，各ステップを説明する．

(1) 検出ステップ 文末述語は，日本語文末述語解析器 Panzer [31] と日本語形態素解析器 JUMAN++ [32] [*7]を用いて解析し，前述の基準に従い難解な文末述語を検出する．Panzer で文末述語範囲の同定と構成要素への分解（機能表現検出）を行い，JUMAN++で助詞の同定を行う．

固有名詞や複合語は，平易にすることが難しい重要な概念を表現していることが多い．したがって，内容部に固有名詞や複合語が含まれていれば，言い換え対象から除外する．固有名詞，複合語の検出には JUMAN++を使用する．

例では，「を配置している」が文末述語として同定され，内容部の動詞「配置する」が基準外の語であるため，難解な文末述語と判定される．

(2) 生成ステップ 検出された難解な文末述語に対して，BERT ベース手法で平易な候補を生成する（手法の詳細は 3.2 節で説明する）．たとえば，「がある」,「がない」,「が多い」,「が並ぶ」,「が広がる」,「を持つ」,「を有する」,「を備える」,「を設置」,「を設ける」といった置換候補が BERT によって生成される．なお実際に生成される候補は「1 階には、半円形の窓がある。」などの完全な文の形であるが，以下では簡潔に表現するため変化の生じる述語部分のみを示す．

生成された候補は正規化され，日本語文生成ライブラリ HaoriBricks3 [33] を使用し元の文末述語の主要機能表現[*8]が再構築される．HaoriBricks3 では，ブリックコードと呼ぶ Ruby コードで，どのような日本語文を合成するかを簡潔に記述することができる．本研究では，タ形・ている形・れる/られる・丁寧・否定の 5 つの機能表現を再構築する[*9]．元の文末述語は「ている形」を持つため，上記の候補例は，「がある」,「がない」,「が多い」,「が並んでいる」,「が広がっている」,「を持っている」,「を有している」,「を備えている」,「を設置している」,「を設けている」に変換される．

(3) 検証ステップ/(4) ランキングステップ 生成された候補が平易であるかを検証し，前述の文末述語の難易度基準を満たさない候補を除外する．例では，「を有している」,「を設置している」,「を設けている」が，難解な候補として除かれる．そして，BERT 尤度・コサイン類似度・言語モデルスコアの 3 つの素性を用いて，残った候補をランキングする（詳細は 3.3 節で説明する）．例では，上から「がある」,「を持っている」,「を備えている」,「がない」,「が多い」,「が並んでいる」,「が広がっている」の順に並べ替えられる．

3.2 言い換え候補の生成手法

文脈を考慮しながらマスクされた単語を予測する能力を持つ BERT を利用する．1 単語ごとの平易化に BERT を適用した Zhou らの研究 [11] を拡張し，複数単語をまとめて言い換える手法を提案する．BERT の入力は，トークン，セグメント，ポジション埋め込みの和で構成される．説明文においては，文末述語は主要な意味を持たないことが多いため，マスクした部分の情報を完全に落としてもある程度復元できると予想できるが，より入力に近い意味の語を得るために，マスクした部分のトークンの埋め込みを平均して保持する（以下，**平均トークン埋め込み**と呼ぶ）．しかし，入力情報を保持しすぎると，マスクした部分と同じ表現や近すぎる表現ばかり生成される可能性が高い．そこで，幅広い出力を得るために，先行研究 [11] を参考にして，入力埋め込みの一部をランダムにゼロにする**ドロップアウト**を導入する．これにより，元の述語と生成される語の関連が弱まり，出力の多様性を確保でき，結果として平

[*6] これらの表現は，不必要に文構造を複雑にすることが多いため，事前にまとめて判定詞「だ」に正規化する．
[*7] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN++
[*8] 原文の文末述語の機能表現はすべて Panzer で取得する．
[*9] これらは旧 JLPT の基準においても 3〜4 級の範囲に収まる．

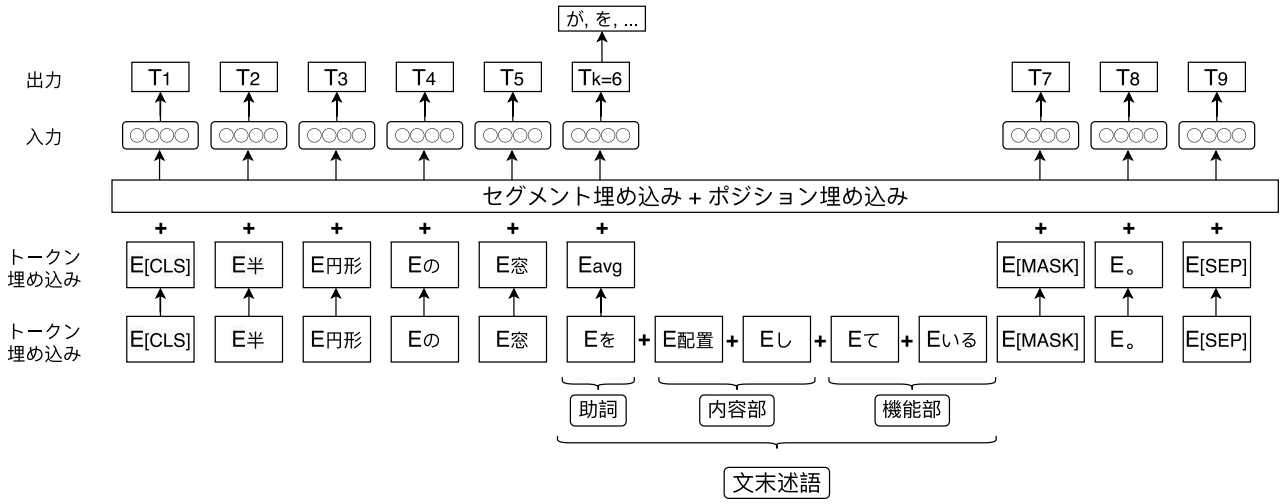

図 3 平均トークン埋め込みを使用した際の助詞生成（[CLS] はトークンの最初，[SEP] はトークンの最後を表す BERT の特殊トークン）

Fig. 3 Generation of a particle by BERT with average token embedding. [CLS] and [SEP] are two special tokens in BERT; [CLS] is added in front of input tokens and [SEP] is a special separator token.

易な表現も候補に含まれやすくなると考えられる．

図 2 に示すように，本手法では，まず助詞を生成し，続いて内容語を生成する．以下，それぞれについて説明する．

助詞生成 平均トークン埋め込みは助詞を含む元の述語の情報を使用する（**図 3**）．w_k を文末述語の先頭とする文長 L の入力トークンを $\boldsymbol{w} = (w_1, ..., w_k, ..., w_L)$ とする．文末述語の平均トークン埋め込みは，w_k から w_L のトークン埋め込みを使用して，次のように計算できる．

$$E_{avg[particle]} = \frac{\sum_{i=k}^{L} E_{w_i}}{L-k+1}$$

ここで，E_{w_i} は i 番目のトークン w_i におけるトークン埋め込みを意味する．w_{k+1} から w_L までのトークンを $\boldsymbol{w}$ から削除し，特殊トークン [MASK] と句点（。）を文末に挿入する．これにより BERT が $w_k, w_{[MASK]}$ の 2 語で残りの文を完結させようとすることにつながる[*10]．すなわち，$w_{[MASK]}$ は擬似的な内容語として機能し，w_k の位置における助詞の生成確率が上がることが期待できる．以上より，新たな BERT の入力トークンは $\boldsymbol{w} = (w_1, ..., w_k, w_{[MASK]}, w_。)$ となる．さらに，トークン埋め込み E_k を $E_{avg[particle]}$ に置き換える．

生成された助詞は，その後に続く表現の生成に大きく影響を与える．幅広い候補生成を行うために，この段階で複数の助詞を生成することが重要である．我々の提案プロセスでは，10 個の出力を生成し，BERT 尤度の高い順に最大 2 つの助詞を保持する．

内容語生成 内容語を生成する際の平均トークン埋め込みの算出方法を**図 4** に示す．直前のステップで生成された助詞を w'_k とすると，入力トークンは $\boldsymbol{w} = (w_1, ..., w'_k, ..., w_L)$ と表すことができる．助詞生成のステップと同様に，内容語生成時の平均トークン埋め込みの計算方法は次のように計算できる．

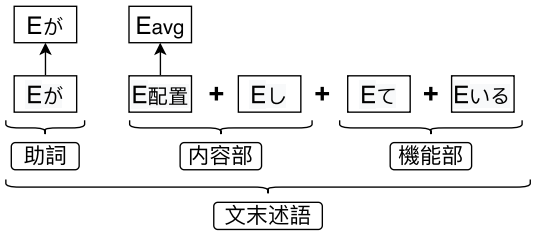

図 4 内容語生成

Fig. 4 Generation of a content word.

$$E_{avg[content]} = \frac{\sum_{i=k+1}^{L} E_{w_i}}{L-k}$$

平均トークン埋め込みで使用したトークンを $\boldsymbol{w}$ から削除し，末尾に句点（。）を加えたものを新たな BERT の入力トークンとし，トークン埋め込みの際に上記の平均トークン埋め込みを使用する．

本研究では，生成された助詞それぞれに対して，BERT 尤度の高い順に 5 つの出力を生成する．3.1 節で述べたように，得られた文は正規化および機能語の復元が行われ，難解な候補は除外される．なお稀に，非文法的な出力が生成されることもあるが，それらの処理はランキングステップで行う．

3.3 候補のランキング手法

このステップでは，最大 10 個の言い換え候補に対してラ

[*10] したがって，入力文の句点の有無によらず，本生成機構においては句点を挿入する．なお，入力文によっては，末尾に感嘆符（！）など別の記号が使用されていることもありえるが，今回は一律に句点のみの挿入とした．

ンキングを行う．文の流暢性（表現が自然であるか）や言い換えの妥当性（意味が保持されているか）をとらえると考えられる次の 3 つの素性ごとにランキングを行い，それらの平均ランクを最終的なランキング結果とする．同ランクの候補があった場合，BERT 尤度が高い方を上位とする．

BERT 尤度 助詞生成および内容語生成時に得られる BERT 尤度の平均値を使用する．この素性は，どれだけ文脈にあった候補が生成できているか，どれだけ元の述語と意味が類似しているか，どれだけ日本語の表現として自然であるか，といった候補の良さを測る最も基本的な指標である．

コサイン類似度 言い換え前後の意味の類似度を測るため，元の述語の内容部と生成された内容語の間の埋め込みベクトルのコサイン類似度を計算する．複数語で構成されている場合，埋め込みベクトルの平均を使用する．各語の埋め込みベクトルを得るため，Common Crawl と Wikipedia で学習された既存の日本語事前学習単語ベクトル fastText [34] [*11] を使用した．

言語モデルスコア 日本語の流暢性を測る指標として，言語モデルの perplexity を使用する．文長 L の入力単語列 w_i^L に対する perplexity（PP）は次式のように計算される．

$$PP = \prod_{i=1}^{L} P(w_i|w_{<i})^{-\frac{1}{L}}$$

日本語 Wikipedia [*12] で学習した Transformer ベースの言語モデル [35] を構築した．日本語のトークン分割には，MeCab [36] [*13] と sentencepiece [37] [*14] を用い，言語モデルの実装には，fairseq toolkit [38] を使用した [*15]．

4. 実験設定

4.1 候補生成手法

4.1.1 提案手法

平均トークン埋め込みとドロップアウトの両方を使用した **BERT[Avg+dp]**，平均トークン埋め込みのみを使用した **BERT[Avg]**，平均トークン埋め込みの代わりに [MASK] トークンを使用した **BERT[MASK]** の 3 つの BERT ベース手法を用いる．トークン分割に MeCab+WordPiece を用いた日本語 BERT モデル [*16] を使用し，ドロップアウト率は Zhou ら [11] に従い 0.3 に設定した．

4.1.2 ベースライン手法

ベースライン手法として，下記の 5 つの手法を実装した（各見出しの括弧内に参照用のキーワードを示す）．既存の言い換え辞書を利用したもの，汎用の学習済モデルを利用したものなど，主要な手法や最新の手法をカバーしている．いずれの手法も，平易語を直接取得するのではなく，複数獲得した類義語をランキングするものであり，提案手法と基本的なプロセスは共通する．しかし，単語単位の置き換えを前提としている点で，提案手法とは大きく異なる．本研究では，生成ステップの有効性を検証することに主眼を置くため，ランキング手法は 3.3 節で提案したものに統一し，図 2 に示す生成ステップ中の内容語生成モジュールのみを，各ベースライン手法で置き換える．いずれの手法も助詞や機能語は置換の対象ではないため，内容語のみの候補獲得を行う．また，BERT を用いたベースライン手法以外は，文脈に依存した機構を持たないため，3.1 節で示した機能語の再構築も行うことで，提案手法となるべく条件を揃える．

シソーラスを用いた手法（シソーラス） 人手で構築された既存のシソーラスから類義語を獲得する手法である．類義語を取得するために，日本語 WordNet 同義対データベース [39] [*17]，分類語彙表 [40] [*18]，内容語換言辞書 [41] [*19] の 3 つのシソーラスを使用した [*20]．これらのシソーラスには，類似度などのスコアは付与されていないため，この段階で候補の絞り込みは行わない．

PPDB を用いた手法（PPDB） 大規模な日本語言い換えデータベース PPDB: Japanese [42] [*21] から類義語を獲得する手法である．PPDB: Japanese の 10best のデータセット [*22] を対象に，内容語の言い換え対を獲得した．PPDB では，言い換え対（それぞれ j, j' とする）に対して，両方向の言い換え確率，すなわち $P(j|j')$ および $P(j'|j)$ が付与されている．たかだか 10 件の候補を生成する提案手法と条件を揃えるため，両確率の平均値が高い順に 10 件を候補とする．

Word2vec を用いた手法（word2vec） 分散表現の類似度に基づく手法である [9], [27]．MeCab でトークン分割した Wikipedia のテキスト全文を用いて学習した 200 次元の CBOW の word2vec モデルを使用した [*23]．対象語とのコサイン類似度が高い順に 10 件を候補とする．

BERT を用いた単語単位の手法 1（BERT-Single1）

[*11] https://fasttext.cc/docs/en/crawl-vectors.html
[*12] https://dumps.wikimedia.org/（2020 年 5 月版のテキストデータ）
[*13] http://taku910.github.io/mecab/
[*14] https://github.com/google/sentencepiece
[*15] https://github.com/pytorch/fairseq デフォルトのハイパーパラメータを用いた．
[*16] https://github.com/cl-tohoku/BERT-japanese（BERT-base_mecab-ipadic-bpe-32k_whole-word-mask）
[*17] http://compling.hss.ntu.edu.sg/wnja/
[*18] https://github.com/masayu-a/WLSP
[*19] http://www.jnlp.org/SNOW/D2
[*20] 使用した類義語対の数は，日本語 WordNet 同義対データベース 11,753 対，分類語彙表 101,070 対，内容語換言辞書 29,639 対である．
[*21] https://ahcweb01.naist.jp/resource/jppdb/
[*22] 10best のデータセットは，15,023,796 件の言い換え対が収録された大規模なものであるが，複数語からなる表現や言い換え関係としては必ずしも妥当でないものも多く収録されており，本研究の用途に適した表現に限定すると大幅にサイズは小さくなる．
[*23] Word2vec モデルの実装には，Python ライブラリの gensim（https://radimrehurek.com/gensim/models/word2vec.html）を用いた．

表 1 人手評価の指標
Table 1 Human evaluation metrics.

	流暢性	妥当性
1	文法的に正しい	文の主要な意味が保持されている
2	少し違和感がある	文脈によっては文の主要な意味が保持されている
3	文法的に正しくない	文の主要な意味が保持されていない

表 2 難解文 525 文に対する平易化ステップごとの候補数に関する統計
Table 2 General statistics of the number of candidates in each simplification step for the targeted 525 difficult sentences.

手法	候補件数			1 つ以上の候補を生成できた文数		
	生成後	検証後	検証後/生成後	生成後	検証後	検証後/生成後
シソーラス	2,552	641	0.251	525	294	0.560
PPDB	2,789	1,431	0.513	523	448	0.857
Word2vec	4,341	2,011	0.463	525	505	0.962
BERT-Single1	4,750	1,649	0.347	525	377	0.718
BERT-Single2	4,746	1,716	0.362	525	378	0.720
BERT[MASK]	4,604	3,053	0.663	525	521	0.992
BERT[Avg]	4,466	2,927	0.655	525	520	0.990
BERT[Avg+dp]	4,521	2,954	0.653	525	521	0.992

対象語の BERT 埋め込みベクトルに対してドロップアウトを適用したうえで，代替語を予測させる Zhou ら [11] の手法を日本語で実装する．提案手法と同じ日本語 BERT モデルを用いて，ドロップアウト率は Zhou ら [11] に従い 0.3 に設定した．提案手法の BERT[Avg+dp] と近いが，上述のように，ベースライン手法はあくまで単語単位の置き換えである．BERT の尤度が高い順に 10 件を候補とする．

BERT を用いた単語単位の手法 2（BERT-Single2） 元の入力文と対象語をマスクした入力文を連結させ，マスクした箇所を予測させる Qiang ら [10] の手法を日本語で実装する．提案手法と同じ日本語 BERT モデルを用いた．BERT の尤度が高い順に 10 件を候補とする．

4.2 評価手法

日本語述語平易化向けの大規模な評価データセットは存在しない．そこで，(1) 出力候補の人手評価および (2) 人手で構築した比較的小規模な参照文との比較を行う．テキストドメインとして，Wikipedia[*24]，ニュース記事[*25]，文化財説明文[*26]の 3 つの説明文を対象とした．それぞれのドメインからランダムに 500 文，合計 1,500 文を抽出し，評価データとした．

(1) 人手評価のために，日本語テキストの校正やアノテーション作業の経験を有する 4 人の日本語母語話者に作業を依頼した．各手法によって生成された出力文それぞれに 2 人の作業者を割り当てた．生成された各候補は，**表 1** に示す流暢性と妥当性の 2 つの指標に基づき作業者に評価してもらった．

(2) 人手参照文の作成のために，1 人の作業者に文末述語を平易に言い換える作業を依頼した．作業者は，日本語テキストの校正や作成の経験だけでなく，小中高の教育現場に関する知見も有している．作業者には，原文および述語を除いた原文を提示し，最大 3 つ程度の平易な言い換えを作成するよう指示した[*27]．内容語については，3 章冒頭で示した難易度基準を満たす語を使用する制約を設け，作業者には使用可能語のリストを渡した．助詞と機能語については，作業の複雑化を避けるため制約を設けなかった．

上記 (1)，(2) の作業者には，事前に十分なインストラクションを与え，また作業者や作業管理者からの作業遂行上の疑問点の解消にも努めた．

5. 実験結果

5.1 検出・生成・検証ステップの結果概要

図 2 の提案プロセスにより，評価データ 1,500 文に対して平易化を行った．まず，検出ステップで 704 文が難解であると判定された．その内 179 文は内容部に複合語もしくは固有名詞を含んでいため除外された[*28]．つまり，残りの 525 文が平易化すべき難解な述語を含む文と判定された．

525 文に対する手法ごとの生成ステップ後および検証ステップ後の候補件数の統計を，**表 2** に示す．生成後の候補件数は，シソーラスと PPDB に比べて，その他の手法が

[*24] https://dumps.wikimedia.org/
[*25] https://www3.nhk.or.jp/news/
[*26] 日本の文化財に関するウェブサイトやパンフレットから収集したデータ [6] である．
[*27] 筆者らの予備的な言い換え試行の経験から，最大 3 つ程度で主要な言い換えをカバーできると判断した．
[*28] 既存の文化財用語辞典 [43] に含まれる専門性の高い用語は，固有名詞として扱った．

表 3 人手評価の作業者間一致度（重み付きカッパ係数）
Table 3 Inter-rater agreement (Weighted Cohen's κ).

セット	文数	流暢性	妥当性
A	5,144	0.757	0.719
B	4,790	0.839	0.792

大幅に上回る．この差は，事前に言い換え語のリストを用意するシソーラスや PPDB と異なり，word2vec や BERT の各手法は類似度や尤度の順に多くの候補を出し続けられることに起因する．また難解な候補を削除した検証後の候補の件数は，提案手法がいずれも約 3,000 件と他の手法を大きく上回る．生成した候補における平易な候補の割合も 65%以上と高い．提案手法は，平易な候補を出す能力についても優れていることが分かる．興味深い点として，提案手法と同じ BERT を使用した BERT-Single1 と BERT-Single2 は，平易な候補の割合が 35%前後と低い．両手法ともに 1 語単位の局所的な穴埋めを行っているため，文脈的制約も大きく，元の難解語に近い語が出やすい（すなわち平易な語が出にくい）と考えられる．

また提案手法は，対象文 525 文のほぼすべてに対し，1 つ以上の平易な候補を生成できていることが分かる．一方，シソーラスベース手法は対象文の 44%に対して 1 つも候補を生成できていない．獲得した候補の流暢性と妥当性については次節で検証するが，この時点で提案手法の生成能力の高さと柔軟性が窺える．

5.2 検証ステップ後の候補に対する評価

5.2.1 人手評価結果

全 16,382 文の検証ステップ後の候補から重複を除いた合計 9,934 文を 2 つのセットに分けたうえで[*29]，各セットにつき 2 名の評価者を割り当て，流暢性と妥当性の評価を行った．**表 3** に，作業者間一致度の指標である重み付きカッパ係数 [44] の値を示す．すべてのスコアが 0.7 以上であり，このような言語評価タスクとしては，一定の信頼性が確保された結果であるといえる [45], [46]．

流暢性と妥当性が，2 人の評価者からともに 1 と評価された候補を Good 候補，それぞれ少なくとも 1 人の評価者から 1 と評価された候補を Acceptable 候補とする[*30]．

表 4 に，候補全体に対する評価結果を示す．提案手法によって生成された Good 候補，Acceptable 候補の合計は，ベースライン手法によって生成された候補数より大幅に多い．なお，シソーラス以外の手法については，各文に対して最大 10 件の候補を出力する都合上，ランクの低い候補は Acceptable にも満たないものも多くなるため，候補全体に占める Good や Acceptable 候補の割合は低くなりがちである．ランキングにより妥当な候補を上位に移動させることが重要である．

[*29] 発注手続きや作業者確保の都合上，均等に 2 つのセットに分割できたわけではないが，なるべく両セットの分量が近くなるようにした．
[*30] Acceptable 候補は Good 候補を包含する．

表 4 獲得された候補数
Table 4 Number of obtained candidates.

手法	Good	Acceptable
シソーラス	111/641 (0.173)	183/641 (0.285)
PPDB	161/1,431 (0.113)	252/1,431 (0.176)
Word2vec	283/2,011 (0.141)	447/2,011 (0.222)
BERT-Single1	181/1,649 (0.110)	413/1,649 (0.250)
BERT-Single2	216/1,716 (0.126)	491/1,716 (0.286)
BERT[MASK]	429/3,053 (0.141)	819/3,053 (0.268)
BERT[Avg]	436/2,927 (0.149)	790/2,927 (0.270)
BERT[Avg+dp]	444/2,954 (0.150)	823/2,954 (0.279)

表 5 Good 候補または Acceptable 候補を候補リストに含む対象文数
Table 5 Number of target sentences that have any good or acceptable candidate.

手法	Good	Acceptable
シソーラス	98/525 (0.187)	135/525 (0.257)
PPDB	136/525 (0.259)	185/525 (0.352)
Word2vec	196/525 (0.373)	268/525 (0.510)
BERT-Single1	117/525 (0.223)	211/525 (0.402)
BERT-Single2	134/525 (0.255)	237/525 (0.451)
BERT[MASK]	214/525 (0.408)	351/525 (0.669)
BERT[Avg]	228/525 (0.434)	353/525 (0.672)
BERT[Avg+dp]	236/525 (0.450)	364/525 (0.693)

表 5 は，平易化対象文 525 文に対する評価結果を示す．Acceptable 候補を生成できた割合は，提案手法いずれにおいても，ベースライン手法を大きく上回り，ベースライン中で最高性能を出した word2vec の結果よりも 15%以上高い．BERT を単語単位で使用したベースライン手法も大幅に凌駕している．これらの結果は，文末述語をまとまった単位で言い換える提案手法の有効性を示している．なお，word2vec や BERT など汎用の事前学習モデルを活用する手法は，言い換え語のリストを事前に用意する手法（シソーラス，PPDB）よりも総じて性能が高い結果となった．柔軟な言い換えが求められる平易化タスクでは，複数の類義語辞書や大規模な言い換え資源を活用するだけでは十分でなく，幅広い候補を獲得できるモデルを適切に使用することが重要であると示唆される．

5.2.2 人手参照文との一致度合い

平易化対象の 525 文に対して人手参照文の構築を行った．候補が 3 つ以上作成された文は 164 文，候補が 2 つ作成された文は 261 文，候補が 1 つ作成された文は 91 文，候補が作成されなかった文は 9 文であった．

生成された候補と人手で構築した言い換え表現（人手参照文）との一致度合いについて報告する．言い換えの妥当性の観点からは，特に内容部の一致が重要であるため，完

全一致だけでなく，内容語の一致にも注目する．**表 6** は人手参照文と一致した候補の統計，**表 7** は人手参照文と一致した候補が得られた対象文の統計を示す．

人手評価と同様に，完全一致，内容語一致ともに，提案手法がベースライン手法よりも優れている．しかし，人手評価の Good 候補数と比較すると，人手参照文における完全一致の候補数は少ない．これは，候補と人手参照文の内容語は同一だが助詞や機能表現が異なる事例が多かったためである．また，人手参照文と内容語も異なる Good 候補も少なからず存在した．たとえば，「1988 年から少林武術ショーを行なっており、現在では世界各地で公演**が催されている**」の例では，人が作成した候補は「が行われている」のみだったが，BERT[Avg+dp] は「がある」,「をしている」,「をおこなっている」,「を展開している」の 4 つの Good 候補を生成した．

言い換え候補に対する大規模な人手評価はコストがかかるため，平易化研究の発展のためには，人手参照文を用意することが重要である．実際，英語の平易化研究における評価は，BLEU [47] や SARI [48] などの尺度を用いて，システム出力結果と人手参照文との文字列比較によりシステム性能に関する大まかな傾向を把握する自動評価が主流である．上述のような人手参照文ではカバーしきれない妥当なシステム出力の存在を過小評価しないよう，適切な人手参照文の構築手法や自動評価手法の開発が求められる．

表 6 人手参照文と一致した候補数

Table 6 Number of candidates included in the human references.

手法	完全一致	内容語一致
シソーラス	36/641 (0.056)	85/641 (0.133)
PPDB	76/1,431 (0.053)	117/1,431 (0.082)
Word2vec	108/2,011 (0.054)	167/2,011 (0.083)
BERT-Single1	110/1,649 (0.067)	137/1,649 (0.083)
BERT-Single2	115/1,716 (0.067)	142/1,716 (0.083)
BERT[MASK]	173/3,053 (0.057)	399/3,053 (0.131)
BERT[Avg]	184/2,927 (0.063)	417/2,927 (0.142)
BERT[Avg+dp]	184/2,954 (0.062)	403/2,954 (0.136)

表 7 人手参照文と一致した候補を候補リストに含む対象文数

Table 7 Number of target sentences that have any candidate included in the human references.

手法	完全一致	内容語一致
シソーラス	36/525 (0.069)	77/525 (0.147)
PPDB	70/525 (0.133)	106/525 (0.202)
Word2vec	96/525 (0.183)	145/525 (0.276)
BERT-Single1	98/525 (0.187)	112/525 (0.213)
BERT-Single2	104/525 (0.198)	117/525 (0.223)
BERT[MASK]	149/525 (0.284)	245/525 (0.467)
BERT[Avg]	164/525 (0.312)	260/525 (0.495)
BERT[Avg+dp]	164/525 (0.312)	260/525 (0.495)

5.3 ランキングステップ後の最終候補に対する評価

表 8 は，ランキング素性の組合せごとの人手評価結果を示す．表中の Acceptable@N は，上位 N 個の候補の中に Acceptable 候補が含まれているか否かを示す指標である．Acceptable@1 は候補を 1 つに絞らなければならない完全自動平易化システムを開発するうえで重要な指標であり，Acceptable@5 は言い換え候補のリストを提示する平易化支援システムへの応用において有用な指標である．全体として提案手法がベースライン手法に比べ高い性能を示しており，Acceptable@1 に注目すると，BERT[Avg+dp] において BERT 尤度とコサイン類似度をランキング素性として用いた手法が最も高い評価結果となった．しかし，どのランキング素性の組合せでも，Acceptable@1 の割合は 0.5 に満たない結果となり，さらなる改善の余地が残されている．一方，BERT[Avg+dp] における Acceptable@5 の割合は，7 割近くに達しており，提案手法は人間の書き手に候補を提示するうえでは有用なレベルにあることが示唆される．ベースライン手法中，最も高い性能を出した word2vec 手法は，Acceptable@1 では提案手法に近い結果を出しているが，Acceptable@5 では提案手法を大きく下回る．以

表 8 使用したランキング素性の組合せごとの Acceptable@N (@1，@5) の結果（括弧内は，対象の 525 文に対する比率）

Table 8 Results of Acceptable@N (@1 and @5) and effectiveness of each ranking feature, with the ratio to the 525 target sentences given in parentheses.

手法	BERT, COS, LM		BERT, COS		BERT, LM		BERT	
	@1	@5	@1	@5	@1	@5	@1	@5
シソーラス	122 (0.232)	135 (0.257)	119 (0.227)	134 (0.255)	121 (0.230)	135 (0.257)	119 (0.227)	134 (0.255)
PPDB	144 (0.274)	184 (0.350)	136 (0.259)	185 (0.352)	147 (0.280)	185 (0.352)	138 (0.263)	185 (0.352)
Word2vec	238 (0.453)	268 (0.510)	221 (0.421)	268 (0.510)	233 (0.444)	268 (0.510)	225 (0.429)	267 (0.509)
BERT-Single1	159 (0.303)	209 (0.398)	156 (0.297)	208 (0.396)	145 (0.276)	208 (0.396)	145 (0.276)	206 (0.392)
BERT-Single2	175 (0.333)	231 (0.440)	186 (0.354)	232 (0.442)	170 (0.324)	230 (0.438)	178 (0.339)	232 (0.442)
BERT[MASK]	248 (0.472)	344 (0.655)	246 (0.469)	344 (0.655)	239 (0.455)	346 (0.659)	241 (0.459)	342 (0.651)
BERT[Avg]	245 (0.467)	350 (0.667)	238 (0.453)	349 (0.665)	231 (0.440)	347 (0.661)	219 (0.417)	349 (0.665)
BERT[Avg+dp]	252 (0.480)	357 (0.680)	257 (0.490)	358 (0.682)	248 (0.472)	355 (0.676)	240 (0.457)	356 (0.678)

表 9 提案手法の優位性を示す事例（上位 5 つの候補のみを表示；ボールド体：Acceptable 候補；下線：Good 候補；“?”：違和感がある/非文法的）

Table 9 Example of the advantage of our proposed methods. Top-5 outputs are shown (boldface: acceptable candidates; underline: good candidates; “?”: awkward or ungrammatical candidates).

入力	正面に下屋**が設けられています**。				
シソーラスベース	[生成候補なし]				
PPDB	?がされています	**が作られています**	?が出されています		
Word2vec	**が備えられています**	**が建てられています**	?が敷かれています		
BERT-Single1	**が備えられています**				
BERT-Single2	**が備えられています**				
BERT[MASK]	**があります**	**を備えています**	?を持っています	?を置かれています	**が付いています**
BERT[Avg]	**があります**	**を備えています**	?を持っています	?をもっています	**が付いています**
BERT[Avg+dp]	**があります**	**を備えています**	?を持っています	?が入っています	**が付いています**

上より，提案手法は，文末述語を対象とした平易化支援システムへの応用における，現時点で最も有望な選択肢であることが示された．

なお，Acceptable@5 では，どのランキング素性の組合せを用いても，結果に大きな差が出ていない．上位 5 件以内に妥当な候補を含めるためには，コサイン類似度や言語モデルスコアに関する計算コストをかけずに BERT 尤度のみを用いるだけで十分であることが分かる．

表 9 に，生成された候補例を示す．シソーラスベース手法は 1 つも候補を生成できていないが，その他の手法は 1 つ以上の Good 候補を生成できている．提案手法は上位 5 件中 Acceptable 候補を 3 件（内 Good 候補を 2 件）生成できており，Good 候補の「があります」がトップにランクインしている．生成された候補「があります」と元の述語「が設けられています」は，それ自体は同義ではないが，文全体を考慮したときに主要な意味は保たれている．1 章で述べた，述語を除いた文脈のみを参照して取り除かれた述語を予測する人間の直感を，マスク言語モデルである BERT がうまく再現し，柔軟な言い換え候補を生成できていることが窺える．一方，文脈を考慮しない PPDB，word2vec，単語単位の局所的な置き換えを行う BERT-Single1，BERT-Single2 では，そのような候補を生成できない．

6. 詳細分析

6.1 各手法の比較

表 5 で提示した少なくとも 1 つ Acceptable 候補が得られた文に焦点を当て，実験手法の全 28 通りの組合せに対してマクネマー検定を行った．**表 10** に，検定に使用した分割表および結果の p 値を示す．

提案手法とベースライン手法の間の全 15 通りの組合せにおいて，p 値は 0.001 以下となり，提案手法の優位性が示された．たとえば，ベースライン手法で最も良い結果を出した word2vec と BERT[Avg+dp] を比べると，対象 525 文中，word2vec のみが Acceptable 候補を生成できた文数が 49 であるのに対し，BERT[Avg+dp] のみが Acceptable 候補を生成できた文数は 145 と大幅に上回る．

続いて，提案手法で導入した，平均トークン埋め込みとドロップアウトの有効性をそれぞれ検証する．BERT[MASK]–BERT[Avg+dp] 間，BERT[Avg]–BERT[Avg+dp] 間の p 値は，それぞれ 0.085，0.052 だった一方，BERT[MASK]–BERT[Avg] 間の p 値は 0.902 だった．これは，平均トークン埋め込みとドロップアウトの組合せが今回の平易化タスクにとって特に有効であることを示している．

表 11 は，文脈のみから推定しにくい述語「が増加する」に対して，元の述語の情報を使用しない BERT[MASK] が Good 候補を生成できていない例を示す．このような場合，元の述語の情報を集約して使用する平均トークン埋め込みが性能の向上に寄与し，BERT[Avg] では上位 5 件中 4 件，Good 候補を生成できている．

表 12 は，平均トークン埋め込みとドロップアウトの組み合わせが有効である例を示す．元の述語の情報を使用しない BERT[MASK] は Acceptable 候補を 1 つも生成できていないが，平均トークン埋め込みを使用する他の 2 つのモデルは，Acceptable 候補を生成できている．そのうえ，BERT[Avg+dp] のみが Good 候補の生成に成功している．対象となるトークンを部分的に符号化する BERT[Avg+dp] は，元の表現と緩やかに類似した候補を幅広く生成することに成功している．

6.2 候補の平易度

表 13 は，元の述語と獲得された候補（表 4 参照）の内容語の難易度変化を示す．3.1 節で述べたように難解な候補は検証ステップで除かれるため，すべての出力候補は最低限の平易化は達成できている．PPDB を除くベースライン手法によって獲得された候補の難易度は，許容難易度の中で最も難しい 2 級が多い一方，提案手法で獲得された候

表 10 平易化手法の各組み合わせに対する人手評価（候補リストにおける Acceptable 候補の有無）の分割表（左上：両手法ともに候補有；左下：列見出しの手法のみ候補有；右上：行見出しの手法のみ候補有；右下：両手法ともに候補無）とマクネマー検定の p 値

Table 10 Contingency tables of human evaluation for each combination of simplification methods and p-values of McNemar's test (upper left in a contingency tables: the number of sentences in which both methods generated any acceptable candidate; lower left: only the method in column header; upper right: only the method in row header; lower right: neither method).

	シソーラス	**PPDB**	**Word2vec**	**BERT-Single1**	**BERT-Single2**	**BERT [MASK]**	**BERT [Avg]**	**BERT [Avg+dp]**
シソーラス	-	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
PPDB	66 | 119 69 | 271	-	<0.001	0.095	<0.001	<0.001	<0.001	<0.001
Word2vec	86 | 182 49 | 208	140 | 128 45 | 212	-	<0.001	0.050	<0.001	<0.001	<0.001
BERT-Single1	52 | 159 83 | 231	86 | 125 99 | 215	122 | 89 146 | 168	-	<0.001	<0.001	<0.001	<0.001
BERT-Single2	60 | 177 75 | 213	94 | 143 91 | 197	135 | 102 133 | 155	205 | 32 6 | 282	-	<0.001	<0.001	<0.001
BERT[MASK]	95 | 256 40 | 134	148 | 203 37 | 137	207 | 144 61 | 113	153 | 198 58 | 116	170 | 181 67 | 107	-	0.902	0.085
BERT[Avg]	95 | 258 40 | 132	153 | 200 32 | 140	211 | 142 57 | 115	158 | 195 53 | 119	174 | 179 63 | 109	319 | 34 32 | 140	-	0.052
BERT[Avg+dp]	98 | 266 37 | 124	156 | 208 29 | 132	219 | 145 49 | 112	163 | 201 48 | 113	180 | 184 57 | 104	333 | 31 18 | 143	345 | 19 8 | 153	-

表 11 平均トークン埋め込みの有効性を示す事例（上位 5 つの候補のみを表示；ボールド体：Acceptable 候補；下線：Good 候補）

Table 11 Example of the effectiveness of average token embedding and dropout (boldface: acceptable candidates; underline: good candidates).

入力	ハムストリングスなどの筋量**が増加する**				
BERT[MASK]	を測定する	を示す	を測る	を表す	を用いる
BERT[Avg]	が多い	**が増える**	**を増やす**	**を高める**	**が増す**

表 12 平均トークン埋め込みとドロップアウトの有効性の例（ボールド体：Acceptable 候補；下線：Good 候補；“?”：違和感がある/非文法的）

Table 12 Example of the effectiveness of average token embedding and dropout (boldface: acceptable candidates; underline: good candidates; “?”: awkward or ungrammatical candidates).

入力	調査開始以来最も多くなる**見通しです。**		
BERT[MASK]	数字です	記録です	
BERT[Avg]	**予定です**	?です	?ます
BERT[Avg+dp]	**予定です**	水準です	**予測です**

補は，最も平易な 4 級が一番多い．候補の平易度に関しても，提案手法はベースライン手法よりも全体的に優れていることが示された．

表 9 の例では，BERT[Avg+dp] は生成ステップにおいて 10 個の候補を生成し，検証ステップにおいて 2 つの候補が除外された．残った候補における内容語のうち，「ある」，「もつ」，「持つ」，「入る」は 4 級の語彙に含まれる．PPDB についても，「する」，「作る」，「出す」と 4 級の内容語が出力されている．それに対し，シソーラス手法は，生成ステップで 11 個の候補を生成したが，いずれも難解語を含んでおり，検証ステップですべて除かれた．また，その他のベースライン手法は，2 級以下の内容語を生成できているものの，4 級の内容語は含まない．

5.1 節で指摘したように，Wikipedia などの一般的なテキストを用いて学習した BERT などの汎用モデルを使用しても，使い方次第では平易な語の出しやすさが変わることが示された．とりわけ，提案手法のように，文末述語を一括して処理することで，BERT の穴埋めタスクにおける文脈的な制約を緩め，単語単位では必ずしも同義とはいえ

表 13 原文と出力候補における内容語の難易度
Table 13 Content word's difficulty of original predicates and generated candidates.

	4 級	3 級	2 級	難解
オリジナル	38 (0.072)	21 (0.040)	58 (0.110)	408 (0.777)
シソーラス	217 (0.339)	89 (0.139)	**335 (0.523)**	0
PPDB	**692 (0.484)**	250 (0.175)	489 (0.342)	0
Word2vec	564 (0.280)	419 (0.208)	**1,028 (0.511)**	0
BERT-Single1	506 (0.307)	253 (0.153)	**890 (0.540)**	0
BERT-Single2	475 (0.277)	311 (0.181)	**930 (0.542)**	0
BERT[MASK]	**1,367 (0.448)**	464 (0.152)	1,222 (0.400)	0
BERT[Avg]	**1,398 (0.478)**	416 (0.142)	1,113 (0.380)	0
BERT[Avg+dp]	**1,363 (0.461)**	426 (0.144)	1,165 (0.394)	0

表 14 各ドメインに対する Acceptable 候補を持つ文数（括弧内は，対象文に対する比率）
Table 14 Number of target sentences that have any acceptable candidate for each domain, with the ratio to the target sentences given in parentheses.

	Wikipedia	ニュース	文化財説明文
シソーラス	**53/184 (0.288)**	47/178 (0.264)	35/163 (0.215)
PPDB	**78/184 (0.424)**	51/178 (0.287)	56/163 (0.344)
Word2vec	97/184 (0.527)	**109/178 (0.612)**	62/163 (0.380)
BERT-Single1	**88/184 (0.478)**	63/178 (0.354)	60/163 (0.368)
BERT-Single2	**100/184 (0.543)**	70/178 (0.393)	67/163 (0.411)
BERT[MASK]	**139/184 (0.755)**	112/178 (0.629)	100/163 (0.613)
BERT[Avg]	**143/184 (0.777)**	107/178 (0.601)	103/163 (0.632)
BERT[Avg+dp]	**147/184 (0.799)**	114/178 (0.640)	103/163 (0.632)

ない多様な候補を生成することが，平易化においては重要であることが示唆される．

6.3 テキストドメイン間の比較

表 14 は，評価に用いた 3 つのテキストドメインそれぞれにつき，Acceptable 候補を生成できた対象文の統計を示す．すべてのドメインで一貫して，提案手法の性能はベースライン手法よりも優れており，特に BERT[Avg+dp] が最も良い結果となった．ドメインによらず，平均トークン埋め込みとドロップアウトの組合せが有効であることを示している．

Word2vec を除く手法において，Wikipedia ドメインでの評価結果が，他のドメインのものよりも優れている点が観察できる．Wikipedia ドメインの平易化タスクが，他のドメインよりも全体的に解きやすいことが示唆される．また，特に BERT を用いた手法すべてにおいて，その傾向が顕著に見られる．本研究で使用した BERT モデルは Wikipedia コーパスで事前学習されたものであるため，他のドメインに比べて Wikipedia テキストにより適合していると考えることもできるだろう．しかし，同じく Wikipedia で学習した word2vec による手法では，ニュースドメインで最も良い結果を出しており，BERT における観察結果とは必ずしも整合しない．学習ドメインと適用先ドメインの関係についてはさらなる検証が必要である．

6.4 エラー分析

提案手法の限界を理解し，さらに性能を向上させる方法を考察するため，どの提案手法でも Acceptable 候補を生成できなかった事例を分析した．言い換え対象 525 文中，そのような事例は合計 137 件あり，それらのエラーの要因を手作業でボトムアップに分類した．**表 15** に，構築したエラーカテゴリーを示す．

述語範囲の誤検出および固有名詞/複合語の誤検出は，検出ステップ（図 2 参照）に関連している．前者のエラーカテゴリーは，特にイディオム表現に起因する．表 15 に示すように，「異彩を放っています」は 1 つの単位として扱うべきイディオム表現であるが，我々の検出ステップでは，この表現を分割し，「を放っています」のみを述語として同定した．しかし，イディオムの一部分のみを言い換えることは不可能である．より正確な言い換え範囲の同定を行うために，イディオム辞書を活用する必要がある．後者のエラーカテゴリーは，上流の言語処理ツールのエラーや辞書の不足に起因するものである．したがって，前処理性能の限界を補うために，固有名詞/複合語の自動検出の仕組みを別途組み込むことが課題である．

生成ステップに関連して，3 つのエラーカテゴリーを同定した．1 つ目は，特に内容語を生成する能力の不足に関するものである．提案手法の枠組みで理論的には候補を生

表 15 Acceptable 候補が生成されなかった文のエラー分類
Table 15 Error category for cases in which no acceptable candidate was generated.

プロセス	エラーカテゴリー	件数	入力例（太字はシステムが検出した文末述語部）	文末述語の人手平易化結果
検出	述語範囲の誤検出	2	戸口を設けるなど異彩を**放っています**。	[述語範囲誤りのため人手平易化できず]
	固有名詞/複合語の誤検出	5	安祥寺から移して**本尊としました**。	[専門用語のため平易化対象外]
生成	モデルの生成能力不足	99	風船を空に向かって**放ちました**。	飛ばしました
	機能部再構築の失敗	2	1 人 30 グラム**まで所持することができます**。	まで持つことができます，持てます
	構造的に不可能	29	二人の縁に結んで**名付ける**。	名前をつける

成できるものの[*31]，BERT モデルが良い候補を生成できなかったエラーを含む．この課題に対処するためには，他のモデルの使用や句レベルの大規模パラフレーズ辞書の導入などの方法が考えられる．2 つ目は，機能部の再構築プロセスの不備に起因している．3.1 節で説明したように，提案手法は平易な 5 つの機能表現を再構築するが，これらを拡充する必要があるかもしれない．しかし，機能部の過剰な再構築は文を複雑にする可能性があるため，追加する機能表現については慎重に考える必要がある．3 つ目は，生成機構上，適切な候補を生成できないエラーである．表 15 に例示した述語「名付ける」は「名前を付ける」と書き換えることが可能であるが，これは 2 つの内容語から成り立つ．我々の手法は，内容語を 1 つしか生成できないため，この問題に対処するためには手法を拡張しなければならない．たとえば，生成ステップにおいて，文の最後に句点を挿入せず，BERT が句点を出力するまで語を生成し続ける手法などが考えられる．しかし，生成する語が多くなるほど，それらの組合せは肥大化し，候補の制御やランキングがより困難になる．

7. おわりに

本論文では，説明文を対象とした日本語文末述語平易化のためのプロセスを定義し，特に中核となる平易な言い換え候補の生成機構にマスク言語モデルの BERT を効果的に活用する手法を提案した．本手法は，文中のマスクされた箇所を復元する BERT の強力な生成能力を生かし，文脈を考慮しながら述語全体を包括的に言い換えることができる．また BERT の入力に平均トークン埋め込みとドロップアウトのメカニズムを導入することで，原文の重要な意味を保持しながら，必ずしも元の述語と同義とは限らない柔軟な言い換え候補を生成することができる．これらは単語単位の言い換えに基づく従来の語彙平易化手法の構造的な限界を克服するものである．さらに，事前学習された既存の大規模な言語モデルのみを利用する本手法は，従来の平易化手法とは異なり，対訳コーパスやシソーラスなどの言語資源に依存しないため，追加のコストをかけずに構築できるという利点もある．

[*31] 人手参照文を参考に，人間が内容語を一語のみ使用し候補を作成できるかを基準とした．

説明文を対象とした評価実験の結果では，提案手法は，難解な述語の約 7 割に対して，原文の意味を保持した平易で流暢な候補を生成でき，従来手法の性能を大幅に上回った．加えて，我々は BERT モデルで導入した平均トークン埋め込みとドロップアウトの有効性，生成された候補の平易度，適用先テキストドメインによる性能の違いを調査し，さらに提案手法において適切な候補を生成できなかったエラーの網羅的・体系的な分類を行った．これらの詳細な分析により，提案手法の特徴や優位性を確認できただけでなく，提案手法の改善方針についての示唆も得られた．

今後の研究では，エラー分析の結果をふまえ，イディオム表現辞書の導入や複数の内容語の生成機構の検討を進める．加えて，BERT の能力を一層引き出し，平易かつ妥当な候補をより多く獲得するために，適用先ドメインのテキストや平易なテキストを用いた BERT の追加訓練にも取り組む．また，提案手法は，人間の書き手に言い換え候補を提示する用途としては実用レベルに達しつつあるため，我々が現在開発中の平易化支援システムに組み込む予定である [49]．そして，書き手にとっての候補生成機構の有用性や，最終的な読み手である日本語学習者や子どもにとっての平易化結果の理解容易性を評価・検証する．

なお，今回は説明文のみを対象としたが，将来的には物語文や論証文 [13] など他のテキストタイプへの本手法の適用可能性を調査する．物語文や論証文は，重要な情報が述語に含まれていることも多く，文末述語の平易化はより挑戦的なタスクとなる．

謝辞 本研究は科研費（課題番号：19K20628）および KDDI 財団調査研究助成（課題名：平易な文化財情報を執筆・翻訳する技術）の支援を受けた．人手評価および参照文作成においては，株式会社バオバブにご協力いただいた．

参考文献

[1] De Belder, J. and Moens, M.-F.: Text Simplification for Children, *Proc. SIGIR Workshop on Accessible Search Systems*, pp.19–26 (2010).
[2] Paetzold, G.H. and Specia, L.: Unsupervised Lexical Simplification for Non-native Speakers, *Proc. 30th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, pp.3761–3767 (2016).
[3] 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).
[4] Saggion, H.: *Automatic Text Simplification*, Morgan & Claypool (2017).
[5] 田中英輝，熊野　正，後藤功雄，美野秀弥：やさしい日本語ニュースの制作支援システム，自然言語処理，Vol.25, No.1, pp.81–117 (2018).
[6] Miyata, R. and Tatsumi, M.: Evaluating the Suitability of Human-oriented Text Simplification for Machine Translation, *Proc. 33rd Pacific Asia Conference on Language, Information and Computation*, pp.167–175 (2019).
[7] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, *Workshop Proc. International Conference on Learning Representations*, pp.1–12 (2013).
[8] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.4171–4186 (2019).
[9] Glavaš, G. and Štajner, S.: Simplifying Lexical Simplification: Do We Need Simplified Corpora?, *Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp.63–68 (2015).
[10] Qiang, J., Li, Y., Yi, Z., Yuan, Y. and Wu, X.: Lexical Simplification with Pretrained Encoders, *Proc. 34th AAAI Conference on Artificial Intelligence*, pp.8649–8656 (2020).
[11] Zhou, W., Ge, T., Xu, K., Wei, F. and Zhou, M.: BERT-based Lexical Substitution, *Proc. 57th Annual Meeting of the Association for Computational Linguistics*, pp.3368–3373 (2019).
[12] Kawahara, D., Kurohashi, S. and Hasida, K.: Construction of a Japanese Relevance-tagged Corpus, *Proc. 3rd International Conference on Language Resources and Evaluation*, pp.2008–2013 (2002).
[13] De Beaugrande, R. and Dressler, W.: *Introduction to Text Linguistics*, Longman (1981).
[14] 国際交流基金，日本国際教育協会：日本語能力試験出題基準改訂版，凡人社 (2002).
[15] Kato, T., Miyata, R. and Sato, S.: BERT-based Simplification of Japanese Sentence-ending Predicates in Descriptive Text, *Proc. 13th International Conference on Natural Language Generation*, pp.242–251 (2020).
[16] Wubben, S., van den Bosch, A. and Krahmer, E.: Sentence Simplification by Monolingual Machine Translation, *Proc. 50th Annual Meeting of the Association for Computational Linguistics*, pp.1015–1024 (2012).
[17] Nisioi, S., Štajner, S., Ponzetto, S.P. and Dinu, L.P.: Exploring Neural Text Simplification Models, *Proc. 55th Annual Meeting of the Association for Computational Linguistics*, pp.85–91 (2017).
[18] Kriz, R., Sedoc, J., Apidianaki, M., Zheng, C., Kumar, G., Miltsakaki, E. and Callison-Burch, C.: Complexity-weighted Loss and Diverse Reranking for Sentence Simplification, *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, pp.3137–3147 (2019).
[19] Zhang, X. and Lapata, M.: Sentence Simplification with Deep Reinforcement Learning, *Proc. 2017 Conference on Empirical Methods in Natural Language Processing*, pp.584–594 (2017).
[20] Zhu, Z., Bernhard, D. and Gurevych, I.: A Monolingual Tree-based Translation Model for Sentence Simplification, *Proc. 23rd International Conference on Computational Linguistics*, pp.1353–1361 (2010).
[21] Xu, W., Callison-Burch, C. and Napoles, C.: Problems in Current Text Simplification Research: New Data Can Help, *Trans. Association for Computational Linguistics*, Vol.3, pp.283–297 (2015).
[22] Katsuta, A. and Yamamoto, K.: Crowdsourced Corpus of Sentence Simplification with Core Vocabulary, *Proc. 11th International Conference on Language Resources and Evaluation*, pp.461–466 (2018).
[23] Inaoka, Y. and Yamamoto, K.: Japanese Grammatical Simplification with Simplified Corpus, *Proc. 2019 International Conference on Asian Language Processing*, pp.41–46 (2019).
[24] Thomas, S.R. and Anderson, S.: WordNet-based Lexical Simplification of a Document, *Proc. KONVENS 2012*, pp.80–88 (2012).
[25] Pavlick, E. and Callison-Burch, C.: Simple PPDB: A Paraphrase Database for Simplification, *Proc. 54th Annual Meeting of the Association for Computational Linguistics*, pp.143–148 (2016).
[26] Pavlick, E., Rastogi, P., Ganitkevitch, J., Van Durme, B. and Callison-Burch, C.: PPDB 2.0: Better Paraphrase Ranking, Fine-grained Entailment Relations, Word Embeddings, and Style Classification, *Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp.425–430 (2015).
[27] 梶原智之，西原大貴，小平知範，小町　守：日本語の語彙平易化のための言語資源の整備，自然言語処理，Vol.27, No.4, pp.801–824 (2020).
[28] Kriz, R., Miltsakaki, E., Apidianaki, M. and Callison-Burch, C.: Simplification Using Paraphrases and Context-based Lexical Substitution, *Proc. 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.207–217 (2018).
[29] Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N. and Huang, X.: Pre-trained Models for Natural Language Processing: A Survey, *Science China Technological Sciences*, Vol.63, pp.1872–1897 (2020).
[30] Shardlow, M.: A Survey of Automated Text Simplification, *International Journal of Advanced Computer Science and Applications, Special Issue on Natural Language Processing 2014*, Vol.4, No.1, pp.581–701 (2014).
[31] 佐野正裕，佐藤理史，宮田　玲：文末述語における機能表現検出と文間接続関係推定への応用，言語処理学会第26回年次大会発表論文集，pp.1483–1486 (2020).
[32] Morita, H., Kawahara, D. and Kurohashi, S.: Morphological Analysis for Unsegmented Languages using Recurrent Neural Network Language Model, *Proc. 2015 Conference on Empirical Methods in Natural Language Processing*, pp.2292–2297 (2015).
[33] 佐藤理史：HaoriBricks3：日本語文を合成するためのドメイン特化言語，自然言語処理，Vol.27, No.2, pp.411–444 (2020).
[34] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T.: Enriching Word Vectors with Subword Information, *Trans. Association for Computational Linguistics*, Vol.5, pp.135–146 (2017).
[35] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I.:

Attention Is All You Need, *Advances in Neural Information Processing Systems*, Vol.30, pp.5998–6008 (2017).
[36] Kudo, T. and Matsumoto, Y.: Japanese Dependency Analysis using Cascaded Chunking, *Proc. 6th Conference on Natural Language Learning*, pp.63–69 (2002).
[37] Kudo, T. and Richardson, J.: SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing, *Proc. 2018 Conference on Empirical Methods in Natural Language Processing* (*System Demonstrations*), pp.66–71 (2018).
[38] Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D. and Auli, M.: fairseq: A Fast, Extensible Toolkit for Sequence Modeling, *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics* (*Demonstrations*), pp.48–53 (2019).
[39] Bond, F., Isahara, H., Fujita, S., Uchimoto, K., Kuribayashi, T. and Kanzaki, K.: Enhancing the Japanese WordNet, *Proc. 7th Workshop on Asian Language Resources*, pp.1–8 (2009).
[40] 国立国語研究所：分類語彙表増補改訂版データベース（ver.1.0）(2004).
[41] 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 回年次大会発表論文集，pp.276–279 (2013).
[42] Mizukami, M., Neubig, G., Sakti, S., Toda, T. and Nakamura, S.: Building a Free, General-domain Paraphrase Database for Japanese, *Proc. 17th Oriental Chapter of the International Committee for the Coordination and Standardization of Speech Databases and Assessment Techniques*, pp.129–133 (2014).
[43] 京都府文化財保護基金（編）：文化財用語辞典，淡交社 (1989).
[44] Cohen, J.: Weighted Kappa: Nominal Scale Agreement with Provision for Scaled Disagreement or Partial Credit, *Psychological Bulletin*, Vol.70, No.4, pp.213–220 (1968).
[45] Landis, J.R. and Koch, G.G.: The Measurement of Observer Agreement for Categorical Data, *Biometrics*, Vol.33, No.1, pp.159–174 (1977).
[46] Artstein, R. and Poesio, M.: Inter-coder Agreement for Computational Linguistics, *Computational Linguistics*, Vol.34, No.4, pp.555–596 (2008).
[47] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: BLEU: A Method for Automatic Evaluation of Machine Translation, *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, pp.311–318 (2002).
[48] Xu, W., Napoles, C., Pavlick, E., Chen, Q. and Callison-Burch, C.: Optimizing Statistical Machine Translation for Text Simplification, *Trans. Association for Computational Linguistics*, Vol.4, pp.401–415 (2016).
[49] 加藤汰一，宮田 玲，立見みどり，佐藤理史：文化財説明文を対象とした平易化支援システムの設計と実装，第 34 回人工知能学会全国大会論文集，pp.1–4 (2020).

加藤 汰一

2019 年名古屋大学工学部電気電子情報工学科卒業．2021 年同大学大学院工学研究科修士課程修了．自然言語処理の研究に従事．

宮田 玲 （正会員）

2012 年東京大学教育学部卒業．2014 年同大学大学院教育学研究科修士課程修了．2017 年同博士課程修了．博士（教育学）．2017 年より名古屋大学大学院工学研究科助教．図書館情報学，自然言語処理，翻訳学の研究に従事．

佐藤 理史 （正会員）

1988 年京都大学大学院工学研究科博士後期課程研究指導認定退学．京都大学工学部助手，北陸先端科学技術大学院大学情報科学研究科助教授，京都大学情報学研究科助教授を経て，2005 年より名古屋大学大学院工学研究科教授．博士（工学）．言語処理学会，人工知能学会，日本認知科学会，ACM 各会員．