テキスト平易化における語彙制約に基づく難易度制御

西原 大貴[†]・梶原 智之^{††}・荒瀬 由紀^{†,†††}

本稿では、入力される英文を特定の学年に合わせた難易度に平易化する難易度制御の手法を提案する。提案手法では、文と単語の両方の難易度を考慮することで、入力文を目標の難易度の文へ書き換える。文の難易度は既存手法と同様、テキスト平易化モデルの入力として目標の文の難易度ラベルを加えることで考慮する。単語の難易度を考慮するために、本研究では3種類の手法を提案する。それぞれ、単語分散表現を拡張して素性として単語難易度を考慮する手法、難解な単語を出力しないハードな語彙制約手法、平易な単語を出力しやすくするソフトな語彙制約手法である。評価実験により、ソフトな語彙制約が有効であることを示す。既存手法は文の難易度のみを考慮しており、省略など構文的な平易化には長けるが、難解な単語をしばしば残す。一方で、提案手法は構文と単語の両方の難易度制御を実現できる。

キーワード：テキスト平易化、テキストの難易度制御

Controllable Text Simplification with Lexical Constraint

DAIKI NISHIHARA[†], TOMOYUKI KAJIWARA^{††} and YUKI ARASE^{†,†††}

We propose a method for controlling the sentence complexity in text simplification tasks. Our text simplification method is successful in translating an input sentence into a specific grade level by considering the complexities of both sentences and words. Sentence complexities are considered by adding the target grade level to an input. In contrast, a word complexity is considered by following three different methods: (a) by extending word embedding by adding a grade level as a feature, (b) by imposing a hard constraint that prohibits outputting complex words, and (c) by imposing a soft constraint that encourages outputting of simple words. Results from the experiments indicate that a soft constraint improves the performance of a text simplification task. Although the existing model that considers only the sentence complexity may control the syntactic complexity aspects like omission, it tends to generate words beyond the target complexity. Our method achieves the control of both syntactic structures and lexical complexities.

Key Words: *Text Simplification, Controllable Text Simplification*

[†] 大阪大学大学院情報科学研究科, Graduate School of Information Science and Technology, Osaka University

^{††} 大阪大学データビリティフロンティア機構, Institute for Datability Science, Osaka University

^{†††} 産業技術総合研究所人工知能研究センター, Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology

本論文の内容の一部は、ACL 2019 Student Research Workshop (Nishihara, Kajiwara, and Arase 2019)、情報処理学会第240回自然言語処理研究会(西原, 梶原, 荒瀬 2019)で報告したものである。

表 1 学年に合わせたテキスト平易化の例

学年	例文
12	According to the Pentagon , 152 female troops have been killed <u>while serving in Iraq and Afghanistan</u> .
7	The Pentagon says 152 female troops have been killed while serving in Iraq and Afghanistan.
5	The military says 152 female have died .

1 はじめに

テキスト平易化 (Shardlow 2014) は、ユーザの文章読解支援を目的として、難解なテキストから平易なテキストへ意味を保持しつつ書き換えるタスクである。各ユーザの語学レベルや読解力、認知力、知識などによって求められるテキストの難易度が異なるため、それぞれのユーザに対応した難易度制御が求められている。特に言語学習に関する教育現場では、教師が多くの時間をかけて各学習者向けに手動で平易なテキストを生成している。インプット仮説 (Krashen 1985) によると、学習者の能力よりわずかに高い難易度の教材を用いることで高い学習効果が得られる。一方、過度に高い難易度の教材では学習効果が低くなり、学習意欲の低下を招く要因にもなる。そのため、教師の負担軽減のために自動的なテキストの難易度制御が求められている (Petersen and Ostendorf 2007)。

テキスト平易化の処理は、主に「省略」と「言い換え」によって実現される。表1の例では、米国の12年生(高校3年生)向けのテキストでの **while serving** が5年生(小学5年生)向けのテキストでは省略されている。また、12年生向けのテキストにおける **According to** が7年生(中学1年生)向けのテキストでは **says** に、**Pentagon** が5年生向けのテキストでは **military** へ言い換えられている。個々のユーザに対応した難易度制御へ向けた第一歩として、本研究では「学年」を対象とし、目的の学年に適した難易度へ文を自動で変換する課題に取り組む。

テキスト平易化における多くの先行研究 (Specia 2010; Coster and Kauchak 2011; Wubben, van den Bosch, and Krahmer 2012; Xu, Napoles, Pavlick, Chen, and Callison-Burch 2016; Kajiwarra and Komachi 2016; Nisioi, Štajner, Ponzetto, and Dinu 2017; Zhang and Lapata 2017; Vu, Hu, Munkhdalai, and Yu 2018; Guo, Pasunuru, and Bansal 2018; Zhao, Meng, He, Andi, and Bambang 2018) では、難解な英文と平易な英文からなるパラレルコーパスに基づく同一言語内の機械翻訳によって単純に「難解」から「平易」への難易度変換を行っており、難易度を細かく制御できない。Scarton and Specia (2018) は、11段階の難易度付きパラレルコーパス (Xu, Callison-Burch, and Napoles 2015) を用いて細かな難易度制御に初めて取り組んだ。この手法では、出力文の目標難易度を入力文にラベルとして付与することにより、同じ入力文であっても

目標難易度に応じて書き換え分けるが、単語の難易度については考慮されていない。本研究では、文全体の大域的な難易度に加えて、個々の単語の局所的な難易度も適切に制御するために、文の難易度とともに単語の難易度も考慮する手法を3種類提案する。それぞれ、単語分散表現を拡張する手法、難解な単語の出力を制限するハードな語彙制約手法、平易な単語の出力を促すソフトな語彙制約手法である。いずれの手法も、既存手法では言い換えられなかった難解な単語を積極的に平易化することが期待できる。

上述の難易度付きパラレルコーパスを用いて、提案手法の有効性を検証する。テキスト平易化では BLEU (Papineni, Roukos, Ward, and Zhu 2002) および SARI (Xu et al. 2016) が標準的な評価指標として用いられる。BLEU が出力文と正解文の語句の一致率を評価するのに対して、SARI は入力文・出力文・正解文の3つを比較することで言い換えるべき語句を正しく言い換えたかを評価する。評価実験の結果、ソフトな語彙制約手法は既存手法と比較して BLEU を 1.04 ポイント、SARI を 0.15 ポイント改善した。さらに詳細な分析の結果、提案手法は文全体の難易度だけでなく単語の難易度も上手く制御でき、積極的な書き換えを促進することが明らかとなった。

2 関連研究

2.1 テキスト平易化

テキスト平易化は難解な文から平易な文への同一言語内の翻訳問題と考えられる。これまでの研究では English Wikipedia と Simple English Wikipedia のパラレルコーパス (W-SW) (Zhu, Bernhard, and Gurevych 2010; Coster and Kauchak 2011) 上で、2段階の難易度変換モデルを訓練してきた。初期の研究では統計的機械翻訳に基づく手法 (Specia 2010; Wubben et al. 2012; Xu et al. 2016; Kajiwar and Komachi 2016) が盛んであったが、近年ではニューラル機械翻訳の成功を受け、注意機構に基づく系列変換モデルを用いる手法 (Nisioi et al. 2017; Zhang and Lapata 2017; Vu et al. 2018; Guo et al. 2018; Zhao et al. 2018) が主流である。しかし、いずれもテキストの難易度を「難解」および「平易」の2段階で変換しており、難易度の細かな制御を行っていない。

2.2 テキストの難易度制御

テキスト平易化において W-SW と並ぶ代表的なデータセットとして Newsela¹ (Xu et al. 2015) がある。Newsela は 11 段階の難易度付きパラレルコーパスであるが、Newsela を用いた先行研究の多く (Zhang and Lapata 2017; Vu et al. 2018; Guo et al. 2018; Zhao et al. 2018) は、細か

¹ <https://newsela.com/data/>

な難易度の違いを考慮せず2段階の難易度変換を行っていた。

Scarton and Specia (2018) は Newsela を用いて、初めてテキスト平易化における難易度制御に取り組んだ。この手法では、出力したい目標の難易度を入力文に付与する。具体的には、文の難易度を示すラベル「<難易度>」を入力文の文頭に付加した上で、Nisioi et al. (2017) と同様の標準的な注意機構に基づく系列変換モデルを訓練した。入力文にラベルを付加して出力文を制御する手法は、多言語翻訳 (Johnson, Schuster, Le, Krikun, Wu, Chen, Thorat, Viégas, Wattenberg, Corrado, Hughes, and Dean 2017) やスタイル変換 (Niu, Rao, and Carpuat 2018) などの他の Text-to-Text Generation タスクでも成功している有望なアプローチである。Scarton and Specia (2018) の手法は目標の難易度を考慮しないベースライン (Nisioi et al. 2017) と比べて BLEU (Papineni et al. 2002) および SARI (Xu et al. 2016) の自動評価を改善し、難易度に応じた平易化を実現できることを示した。しかし、この手法では文全体の難易度のみを考慮しているため、省略の操作には優れる一方で、ターゲットユーザにとって難易度の高い単語が言い換えられずに出力文に残る場合が少なくない。そのため本研究では、Scarton and Specia (2018) の手法において、文の難易度に加えて単語の難易度も考慮することで、より正確なテキストの難易度制御を行う。

3 単語の難易度を考慮したテキストの難易度制御

本研究では、Scarton and Specia (2018) の手法において、目標の難易度に適した単語の出力を促すために、単語の難易度を考慮する3種類の手法を提案する。3.1節では、各単語の分散表現を拡張することで、素性として単語の難易度を考慮する手法を提案する。3.2節では、目標の難易度に応じて出力可能な単語を制限するハードな語彙制約を提案する。3.3節では、目標の難易度に応じて各単語を重み付けするソフトな語彙制約を提案する。

3.1 素性としての単語難易度の考慮

単語の難易度を考慮するために、埋め込み層を拡張する。本手法では、難解な単語は難解な文中で出現しやすく平易な単語は平易な文中で出現しやすいと仮定して、文の難易度ごとの各単語の出現確率の分布を用いて埋め込み層を拡張する。

まず、大規模な生コーパス上で事前訓練された k 次元の単語分散表現を用意する。この単語分散表現は、単語の意味を表現するベクトルであり、難易度の情報は持たないと仮定する。これに単語の難易度を表現する n 次元ベクトル $\mathbf{a} = (a_1, a_2, \dots, a_n)$ を結合することで拡張した $k+n$ 次元のベクトルを、符号化器および復号器の埋め込み層として使用する。ここで n は、訓練データにおける文の難易度の種類数であり、本研究では11である。拡張するベクトル $\mathbf{a}$ の各

要素 a_i ($i = 1, 2, \dots, n$) は n 段階の各難易度に対応しており, 次式で定められる.

$$a_i = \frac{P(w | u_i)}{\sum_{j=1}^n P(w | u_j)}, \quad (i = 1, 2, \dots, n). \quad (1)$$

ここで, $P(w | u_i)$ は訓練データの各難易度 u_i における単語 w の出現確率を表している. すなわち, ベクトル $\mathbf{a}$ は, 単語の出現確率の分布として単語の難易度を表現する素性である. 意味ベクトルおよび難易度ベクトルの情報を保持するために, テキスト平易化モデルの訓練時には, これらの埋め込み層は固定する.

類似した手法として, 対話応答生成タスクでは, 復号器の埋め込み層において話者の分散表現を付加するスタイル変換の手法 (Li, Galley, Brockett, Spithourakis, Gao, and Dolan 2016) が提案されている. この手法はスタイルごとに1つの固定された情報を復号器の入力として付加するが, 提案手法では符号化器および復号器の入力として単語ごとに固有の情報を付加する.

3.2 ハードな語彙制約による単語難易度の考慮

目標の難易度に合致した単語を出力するために, 出力語彙を制限する. 本手法では, 訓練データにおいて各単語が出現する文の難易度のうち, 最も平易な難易度をその単語の難易度であると仮定する. 例えば, ある単語が3年生向けテキストおよび4年生向けテキストに出現する場合, その単語の難易度は3とし, 3年生から12年生までの文で出現可能として扱う. つまり, 目標の難易度が3未満の場合にはこの単語を出力しないという制約を復号器に加える.

テキスト平易化の推論では, 各タイムステップの復号器出力を $\mathbf{o} = [o_1, \dots, o_N] \in \mathbb{R}^N$ (N は語彙サイズ) として, 最大要素のインデックス $\arg \max_i o_i$ に対応する単語をモデルが出力する. 出力する単語を目標の難易度 T に適した単語のみに制限するために, まず語彙制約マスク

$$\mathbf{m} = [m_1, \dots, m_N], \quad m_i = \begin{cases} 1 & \text{if } b_i \leq T \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

を用意する. ここで, b_i はインデックス i に対応する単語の難易度を表す. これを用いて, 図1に示すようにモデルの出力を

$$\mathbf{o}' = \mathbf{o} \odot \mathbf{m} \quad (3)$$

と改めることで出力語彙を制限する. $\odot$ はベクトルの要素ごとの積を計算する演算子である.

訓練データから単語の難易度を推定する他に, 既存の単語難易度辞書 (English Vocabulary

図 1 ハードな語彙制約により単語難易度を考慮する提案手法

Profile²や Maddela and Xu (2018) の辞書³など) を語彙制約のために利用することも考えられる。しかし、Newsela コーパスにおける語彙の網羅率が English Vocabulary Profile では 8.87%, Maddela and Xu (2018) の辞書では 26.18% と低く、ほとんどの単語に難易度を割り当てられなかったため、本研究では使用しなかった。

3.3 ソフトな語彙制約による単語難易度の考慮

目標の難易度に適した単語の出力を促すために、図 2 のように損失の計算時に各単語を難易度に応じて重み付けする。この重み $f(w, l)$ は、文の難易度 l に対する単語 w の出現しやすさを表す。本手法では、難解な単語は難解な文中で出現しやすく、平易な単語は平易な文中で出現しやすいと仮定し、各文の難易度における単語の出現頻度を手がかりに難易度に適した単語を推定する。ある単語の損失が大きくなると、その単語の出力を漏らさないように学習が促されるため、損失の重み付けによって目標難易度に適した単語が出力されやすくなると期待できる。

一般に、系列変換モデルの損失はクロスエントロピーを用いて求められる。各タイムステップのモデルの出力を $\mathbf{o} = [o_1, \dots, o_N] \in \mathbb{R}^N$ とすると、損失 $L(\mathbf{o}, \mathbf{y})$ は、

$$L(\mathbf{o}, \mathbf{y}) = -\mathbf{y} \log \mathbf{o}^\top = -\log o_c \quad (4)$$

で得られる。ただし、 $\mathbf{y} = [y_1, \dots, y_N] \in \mathbb{R}^N$ は正解単語を指す c 番目の要素が 1 で他の要素が全て 0 である one-hot ベクトルである。提案手法では、式 (4) を

$$L(\mathbf{o}, \mathbf{y}, w_c, l) = -f(w_c, l) \log o_c \quad (5)$$

と改めることで、目標難易度に応じて損失を重み付けする。ここで、語彙を V とすると $w_c \in V$ は正解単語を表す。重み $f(w, l)$ には、TFIDF (Term Frequency Inverse Document Frequency)

² <http://www.englishprofile.org/wordlists>

³ https://github.com/mounicam/lexical_simplification

図 2 単語 w と目標難易度 l を用いて損失を重み付けするソフトな語彙制約

または PPMI (Positive Pointwise Mutual Information) を利用する.

TFIDF 難易度ごとの単語の重要度を推定するために, 各難易度の文の集合を文書とみなして次式の TFIDF を考える.

$$\text{TFIDF}(w, l) = P(w | l) \log \frac{n}{\text{DF}(w)}. \quad (6)$$

ただし, $P(w | l)$ は特定の難易度 l における単語 w の出現確率, n は難易度の総数⁴, $\text{DF}(w)$ は単語 w が出現する難易度の種類数を表す. そのため TFIDF は, 特定の難易度において特徴的な単語により大きな重みを与える.

PPMI 各文の難易度と単語の共起の強さを推定するために, 次式の自己相互情報量 (PMI, Pointwise Mutual Information) を考える.

$$\text{PMI}(w, l) = \log \frac{P(w, l)}{P(w)P(l)} = \log \frac{P(w | l)}{P(w)}. \quad (7)$$

ただし, $P(w)$ はコーパス全体における単語の w の出現確率, $P(w | l)$ は式 (6) と同じく特定の難易度 l における単語 w の出現確率を表す. ここで, PMI が負数を取る場合, l と w の関連が低いと言える. すなわち, PMI(w, l) の値が小さくなる単語 w は, 難易度に依存せず広く使用される単語であると考えられる. 特定の文の難易度において頻出する単語の出力を促すことが目的であるため, PMI が負となる単語は重み付けの考慮から除外する. そのため, PMI の正のみを利用した次式の PPMI (Positive PMI) を用いる.

$$\text{PPMI}(w, l) = \max(\text{PMI}(w, l), 0). \quad (8)$$

TFIDF および PPMI は, いずれも値域が $[0, \infty)$ である. 難易度に対して偏りのない単語の損失の重みを 1 倍にするために, $f(w, l)$ はバイアスとして 1 を加えて

⁴ 本実験では 2 から 12 までの難易度を扱うため $n = 11$.

$$f(w, l) = \text{Func}(w, l) + 1, \quad \text{Func} \in \{\text{TFIDF}, \text{PPMI}\} \quad (9)$$

とする.

4 実験

4.1 実験設定

4.1.1 データセット

本研究では多段階の難易度制御を行うためのデータセットとして, Scarton and Specia (2018) と同じく Newsela (Xu et al. 2015) を用いる. Newsela の各文書は 1 つのニュース記事に対応し, 専門家によって文書単位で平易化されている. 各文書には, 米国学校制度における学年に対応する難易度が付与されており, 2~12 の 11 段階の難易度を利用できる.

本研究で使用した Newsela は, Xu et al. (2015) によって文アライメントが取られ, Zhang and Lapata (2017) によって訓練用 1,070 文書 (94,208 文対), 検証用 30 文書 (1,129 文対), 評価用 30 文書 (1,077 文対) に分割されたデータセットである. 本実験では, Scarton and Specia (2018) と同じく, 各文の難易度をその文が含まれる文書の難易度とする. なお, Scarton and Specia (2018) は一般に公開されている Newsela (Xu et al. 2015) よりも大規模な非公開のコーパスを利用しているため, この論文で報告されている性能と本研究で実施する実験結果は直接比較できない. そこで本研究では Scarton and Specia (2018) の手法を再現した上で比較を行う.

4.1.2 評価指標

本実験の評価では, テキスト平易化の先行研究 (Xu et al. 2016; Nisioi et al. 2017; Zhang and Lapata 2017; Scarton and Specia 2018) と同じく, BLEU (Papineni et al. 2002) および次式の SARI (Xu et al. 2016) を用いる.

$$\text{SARI} = \frac{1}{3}(F_{\text{add}} + F_{\text{keep}} + P_{\text{del}}). \quad (10)$$

ここで, F_{add} , F_{keep} , P_{del} は, 入力文・出力文・正解文の 3 つを比較して計算される. それぞれ, モデルが入力文に対して正しく追加した n -gram⁵ の F 値, 入力文から正しく保持した n -gram の F 値, 入力文から正しく削除した n -gram の適合率である. BLEU は主に意味と文法の観点で人手評価との相関を持ち, SARI は意味と文法に加えて難易度の観点でも人手評価との相関を持つ (Xu et al. 2016).

評価指標の特性を理解するために, source および reference-shuffle の 2 つのベースラインを考える. source は入力文を編集せずにそのまま出力するものである. reference-shuffle は, 正解文

⁵ 1-gram から 4-gram まで考慮する.

の語順を無作為に入れ替えるものである。4.2 節に示すように、入力を書き換えない source は高い BLEU を得られるが、SARI は低い。一方、reference-shuffle は高い SARI を得られるが、BLEU は低い。つまり、BLEU または SARI の一方のみが高いモデルは容易に得られてしまうため、両方のスコアを同時に改善したモデルが優れていると考えられる。

4.1.3 比較手法

実験項目は以下の通りである。項目 (1) から (4) が比較手法、(5) から (8) が提案手法である。

- (1) **s2s** は、注意機構に基づく単純な系列変換モデルである。
- (2) **s2s+grade** は、入力文頭に目標難易度のラベル⁶を付与して出力文の難易度を制御する既存手法 (Scarton and Specia 2018) に基づくモデルである。
- (3) **s2s (GloVe)** は注意機構に基づく系列変換モデルにおいて、埋め込み層として GloVe (Pennington, Socher, and Manning 2014) の訓練済み単語分散表現を用いるモデルである。
- (4) **s2s (GloVe)+grade** は s2s (GloVe) において (Scarton and Specia 2018) の手法を適用したモデルである。
- (5) **s2s (GloVe+単語難易度ベクトル)+grade** は、Scarton and Specia (2018) の手法において、埋め込み層に提案手法 (3.1 節) を適用したモデルである。本実験では、GloVe の 512 次元の訓練済み単語分散表現に 11 次元の単語難易度ベクトルを追加した 523 次元を埋め込み層として用いた。
- (6) **s2s+grade+hard** は、s2s+grade においてハードな語彙制約を行う提案手法 (3.2 節) である。
- (7) **s2s+grade+TFIDF** は、s2s+grade のベースラインモデルに TFIDF に基づく損失の重み付けを適用した提案手法 (3.3 節) である。各単語の TFIDF 値は訓練データを用いて計算した。
- (8) **s2s+grade+PPMI** は、s2s+grade のベースラインモデルに PPMI に基づく損失の重み付けを適用した提案手法 (3.3 節) である。各単語の PPMI 値は訓練データを用いて計算した。

4.1.4 手法の実装

手法 (3)～(5) で使用する単語分散表現 GloVe は、次元数 512、窓幅 15 として English Wikipedia⁷ および Newsela を用いて訓練した。なお、文の難易度ラベルに対応する分散表現を得るため、Newsela の全ての文頭に各文の難易度を示すラベルを付与して GloVe の訓練を行った。

本実験では、Marian⁸ (Junczys-Dowmunt, Grundkiewicz, Dwojak, Hoang, Heafield, Neckermann,

⁶ 本研究で用いるラベルは具体的に、 $\langle 2 \rangle, \langle 3 \rangle, \dots, \langle 12 \rangle$ の 11 種類である。

⁷ <https://dumps.wikimedia.org/enwiki/20191201/enwiki-20191201-pages-articles-multistream.xml.bz2>

⁸ <https://github.com/marian-nmt/marian/commit/02f4af4>

表 2 実験結果 (太字は s2s+grade と比較して評価値が優れているもの)

	BLEU ↑	SARI ↑
source	21.37	2.82
reference-shuffle	4.22	43.07
(1) s2s	20.43	28.21
(2) s2s+grade	20.82	29.44
(3) s2s (GloVe)	22.47	22.73
(4) s2s (GloVe)+grade	23.05	24.13
(5) s2s (GloVe+単語難易度ベクトル)+grade	23.55	24.58
(6) s2s+grade+hard	13.73	31.65
(7) s2s+grade+TFIDF	21.00	29.58
(8) s2s+grade+PPMI	21.86	29.59

表 3 素性として単語難易度を考慮する提案手法の失敗例

失敗例 1	入力文	<4> It enhances friendships among people,” he said.
	正解文	It also builds friendships.
	出力文	It enhances friendships among people,” he said.
失敗例 2	入力文	<3> Hauanio hired a crew with a trailer .
	正解文	Hauanio hired workers with a trailer .
	出力文	She hired a crew with a trailer .

Seide, Germann, Aji, Bogoychev, Martins, and Birch 2018) を使用してテキスト平易化モデルを構築した. 符号化器および復号器には 2 層の Bi-LSTM を用い, 隠れ層は 1,024 次元で dropout 率 0.2, 埋め込み層は 512 次元で dropout 率 0.1 とした. 埋め込み層は符号化器と復号器および出力層で共有した. 最適化には Adam (Kingma and Ba 2015) を利用し, 検証データにおける perplexity が 8 エポック改善しなくなったところで訓練を終了した. ニューラルネットワークの初期値依存性を考慮し, いずれの実験も初期値を無作為に変更しつつ 3 回ずつ試行し, 得られた評価値の平均を報告する.

4.2 実験結果

4.2.1 素性としての単語難易度を考慮する提案手法 (3.1 節) の実験結果

実験結果を表 2 に示す. 提案手法 (5) s2s (GloVe+単語難易度ベクトル)+grade は, 比較手法 (3) s2s (GloVe) および (4) s2s (GloVe)+grade よりも BLEU および SARI が優れており, 素性としての単語難易度を考慮する提案手法の有効性が確認された. しかし, 訓練済みの単語分散

表 4 ハードな語彙制約により単語難易度を考慮する提案手法の失敗例

失敗例 1	入力文	<4> It took months for the wounds to heal .
	正解文	It took him months to heal .
	出力文	It took months for the seas to talk .
失敗例 2	入力文	<2> Recently , Richie and his class were matching subjects with verbs .
	正解文	Not long ago , Richie and his class were matching nouns with verbs .
	出力文	In the country , the people in the country have a lot of food .

図 3 ハードな語彙制約において各難易度で使用可能な語彙サイズ

表現を用いた手法 (3), (4), (5) は, それを用いない手法 (1) s2s および (2) s2s+grade に比べて SARI が低く, 入力から正しい書き換えを行っていない. モデル出力の失敗例を示した表 3 によると, 失敗例 1 では **enhances** を **builds** に言い換えできていない. 失敗例 2 では, **crew** を **worker** に言い換えできていない. このように, 期待とは異なり難解な単語を残したまま出力する例が見られた. そこで手法 (6)~(8) では, 訓練済みの単語分散表現を用いるのではなく, 分散表現も同時に学習するアプローチを用いた.

4.2.2 ハードな語彙制約により単語難易度を考慮する提案手法 (3.2 節) の実験結果

表 2 によると, 提案手法 (6) s2s+grade+hard は既存手法 (2) s2s+grade と比較して SARI を 2.21 ポイント改善しているが, BLEU が著しく低く正しい単語を出力できていない.

表 4 の失敗例 1 では, 期待通り難解な単語 **wounds** の出力を避けることができたが, 誤って異なる意味の単語 **seas** を出力してしまった. 図 3 に示すように, 各難易度で使用可能な語彙サイ

表 5 ソフトな語彙制約により単語難易度を考慮する提案手法の出力例

学年	例文
入力 12	In its original incarnation during the ‘ 60s , African-American ” freedom songs ” aimed to motivate protesters to march into harm ’s way <u>and , on a broader scale , spread news of the struggle to a mainstream audience .</u>
出力 7	s2s+grade: In the 1960s , African-American ” freedom songs are aimed to motivate protesters to march into harm ’s way . s2s+grade+PPMI: In its original people in the 1960s , African-American ” freedom songs are aimed to inspire protesters to march into harm ’s way .
出力 4	s2s+grade: In the 1960s , African-American ” freedom songs are aimed to motivate protesters to march into harm ’s way . s2s+grade+PPMI: African-American ” freedom songs are aimed to inspire protesters to march into harm ’s way .
正解 4	During the 1960s , there were African-American ” freedom songs . ”

ズを見ると、難易度 2 では 1,059 語彙 (2.82%) の単語に制限されている。このような厳しすぎる制約によって、自然な文を構成する上で必要な単語を利用できず、表 4 の失敗例 2 に示すように入力文とは意味が全く異なる文を出力してしまったと考えられる。

4.2.3 ソフトな語彙制約により単語難易度を考慮する提案手法（3.3 節）の実験結果

表 2 によると、提案手法 (7) s2s+grade+TFIDF および (8) s2s+grade+PPMI は、BLEU と SARI の両方において既存手法である (2) s2s+grade を上回った。特に (8) s2s+grade+PPMI の BLEU と SARI は (2) s2s+grade と比較してそれぞれ 1.04 ポイント (p 値 0.024)、0.15 ポイント改善した。ソフトな語彙制約によって単語難易度を考慮することで、出力文を崩壊させずにより目標難易度に適した単語を出力できたと考えられる。

表 5 にモデルの出力例を示す。(8) s2s+grade+PPMI は **incarnation** を 7 年生向けに **people** へ、**motivate** を 4 年生向けに **inspire** へ書き換えている。いずれのモデルも **and** , **on** ~ の並列構造を省略できているが、(8) s2s+grade+PPMI は 4 年生向けの文をより短い文へ省略できている。

次節では、BLEU と SARI の両方を改善した提案手法 (8) s2s+grade+PPMI に焦点を当てて考察と分析を行う。

表 6 詳細な分析結果

	self-BLEU ↓	文長の平均絶対誤差 ↓	平均 PMI ↑
source	100.0	10.73	0.08
reference	18.30	0.00	0.23
(1) s2s	37.60	4.38	0.12
(2) s2s+grade	31.96	3.77	0.15
(8) s2s+grade+PPMI	31.38	3.69	0.19

5 考察と分析

BLEU および SARI による自動評価において最高性能を達成した (8) s2s+grade+PPMI (提案手法) について, 本節では (2) s2s+grade (既存手法) と比較しながら 3 つの追加の自動評価指標および人手による分析により詳細な検証を行う. 既存のテキスト平易化モデルは, 入力文をあまり書き換えず, 難解な表現を出力文に残してしまいやすい (Zhang and Lapata 2017). そこで 5.1 節では, モデルが入力文を積極的に書き換えているかを評価する. また, テキスト平易化の処理は主に省略と言い換えによって実現される. そのため, 5.2 節では省略による難易度制御の評価を, 5.3 節では言い換えによる難易度制御の評価を行う. 5.4 節では, 難易度の制御に関する詳細な分析のために, 目標難易度ごとに文および単語の難易度による評価を行う. 最後に 5.5 節では, 人手による出力文のエラー分析から今後の改善策を議論する.

5.1 書き換えの積極性の評価

書き換えの積極性を評価するために, self-BLEU (Sun and Zhou 2012) を用いる. 出力文と正解文の一致度を評価する通常の BLEU とは異なり, self-BLEU は入力文と出力文の一致度を評価する. つまり, self-BLEU の値が低いほど, 入力文を積極的に書き換えるモデルであると言える. 表 6 に示した結果のうち, source は入力文を編集せずにそのまま出力文とし, reference は正解文を出力文とみなしたときの評価である. 表 6 によると, self-BLEU においても s2s+grade+PPMI が最も優れており, 目標難易度に適した単語を重視することで入力文に含まれない単語を積極的に出力できていることがわかる. s2s は目標の難易度を考慮しないため, あまり書き換えを行わない (self-BLEU が高い) 保守的なモデルになったと考えられる.

5.2 省略による難易度制御の評価

省略による難易度制御の品質を評価するために, 文長の平均絶対誤差を用いる. 文長の平均絶対誤差は, 出力文の単語数に基づいて次式で得られる.

$$(\text{文長の平均絶対誤差}) = \frac{1}{S} \sum_i |\text{Len}(r_i) - \text{Len}(t_i)|. \quad (11)$$

ここで、 S は評価用データセットの文数、 r_i ($i = 1, \dots, S$) は正解文、 t_i は r_i に対応する出力文、 $\text{Len}(s)$ は文 s の単語数を表す。文長の平均絶対誤差が低いほど、適切な長さの文を生成するモデルであると言える。表 6 を見ると、s2s+grade は s2s から文長の平均絶対誤差が 0.61 ポイント改善され、s2s+grade+PPMI は s2s+grade と同程度である。すなわち、文の難易度を考慮することによって出力文の文長が正解文と近くなり、省略による難易度制御が改善されている。

5.3 言い換えによる難易度制御の評価

言い換えによる難易度制御の品質を評価するために、平均 PMI を用いる。平均 PMI は、次に示す通り出力文に含まれる各単語と目標難易度の一致度に基づいて得られる。

$$(\text{平均 PMI}) = \frac{1}{W} \sum_{s \in \text{Target}} \sum_{w \in s} \text{PMI}(w, l_s). \quad (12)$$

ここで、 W は出力文の集合 Target に含まれる単語数、 l_s は文 s の目標難易度、 w は文 s に含まれる単語である。PMI は訓練データを用いて式 (7) によって求めた。平均 PMI が高いほど、適切な難易度の単語を出力するモデルであると言える。表 6 を見ると、平均 PMI は s2s+grade+PPMI が最も高い。したがって s2s+grade+PPMI は、s2s+grade の省略による難易度制御の品質を保ちつつ、より目標難易度に適した単語を出力できることが明らかになった。

5.4 目標難易度ごとの出力文における文と単語の難易度

難易度の制御についてより詳細な分析を行うために、評価用データセットの各入力文に対して、入力文よりも平易な全ての難易度⁹に向けての平易化を行った。表 7 に、s2s+grade と s2s+grade+PPMI に対する目標難易度ごとの FKGL (Kincaid, Fishburne Jr., Rogers, and Chissom 1975) および平均 PMI を示す。FKGL は、テキストの可読性を評価する自動評価尺度であり、次式で得られる。

$$\text{FKGL} = 0.39 \frac{\# \text{words}}{\# \text{sentence}} + 11.8 \frac{\# \text{syllables}}{\# \text{words}} - 15.59. \quad (13)$$

ここで、 $\# \text{words}$ は単語数、 $\# \text{sentence}$ は文数、 $\# \text{syllables}$ は音節数である。FKGL の評価値は米国学校制度における学年に対応しており、本実験ではこれ为目标難易度と比較することにより各手法の難易度制御の品質を分析する。

FKGL の分析によると、いずれのモデルも過度に平易な文に書き換えてしまっているが、目

⁹ 正解文は各入力文に対して一部の難易度についてのみ付与されているため、ここでは正解文を用いる評価は実施できない。

¹⁰ 本実験で使用したコーパスの正解文には、<9>から<11>の難易度に対応する文が存在しない。

表 7 各難易度における s2s+grade (既存手法) と s2s+grade+PPMI (提案手法) の FKGL と平均 PMI

学年 ¹⁰	FKGL			平均 PMI	
	s2s+grade	s2s+grade+PPMI	差分	s2s+grade	s2s+grade+PPMI
<8>	4.92	5.33	+0.41	0.11	0.12
<7>	4.87	5.25	+0.38	0.10	0.12
<6>	4.47	4.56	+0.09	0.12	0.14
<5>	3.51	3.71	+0.20	0.13	0.15
<4>	2.68	2.69	+0.01	0.16	0.19
<3>	2.06	1.89	-0.17	0.18	0.23
<2>	1.81	1.44	-0.37	0.20	0.24
MAE	1.52	1.45	—	—	—

標学年に対する平均絶対誤差 (MAE) を比較すると s2s+grade+PPMI の方が優れている。表 7 中, FKGL の差分は学年ごとに s2s+grade と s2s+grade+PPMI の FKGL の差を示している。これによると, s2s+grade+PPMI は目標学年が小さい場合はより平易な文を出力し, 目標学年が大きい場合はより難解な文を出力しているため, s2s+grade と比較してより明確に難易度を区別した制御ができています。この結果から, 単語の難易度を考慮することで, テキスト平易化の難易度制御に寄与することが分かる。

平均 PMI の評価によると, s2s+grade+PPMI は全ての学年で s2s+grade より優れている。したがって, 単語の難易度に基づく損失の重み付けを行う提案手法が, 我々の期待通りに目標の学年に適した単語を出力することが確認できた。

5.5 エラー分析と人手評価

今後の課題を明らかにするために, 評価用データセットから無作為抽出した 50 文について, s2s+grade (既存手法) および s2s+grade+PPMI (提案手法) のエラー分析 (5.5.1 節) と人手評価 (5.5.2 節) を行った。

5.5.1 エラー分析

テキスト平易化における代表的な操作である省略および言い換え (置換) について, それぞれの細分類ごとにエラー数を著者の一人が人手で調べた。まず省略に関するエラー分析のために, 入力文・出力文・正解文の三つ組を比較して, 入力文中の省略すべき箇所, 正しく省略した箇所 (省略成功), 省略すべきだが省略しなかった箇所 (省略不足), 出力すべきだが過剰に省略している箇所 (省略過多) を数えた。次に置換に関するエラー分析のために, 入力文中の置換すべき箇所, 正しく置換した箇所 (置換成功), 誤った表現に置換した箇所 (置換失敗), 置換すべきだが置換しなかった箇所 (置換不足), 置換すべきでないが置換した箇所 (置換過多)

表 8 エラー分析の例（出力失敗例）

入力文	<u>She</u> said <u>the college application process</u> can be especially <u>stressful</u> for immigrant students.
正解文	<u>Albrecht</u> said applying for college can be especially tough for immigrants.
s2s+grade	<u>She</u> said the college program can be especially <u>hard</u> for <u>immigrant students</u> .
s2s+grade+PPMI	<u>She</u> said <u>it</u> can be very <u>hard</u> for <u>immigrant students</u> .
入力文	Media reports said <u>at least</u> 10 people were <u>detained</u> in five Turkish cities for organizing the <u>boycott</u> .
正解文	<u>More than</u> 10 people were <u>arrested</u> for organizing the <u>absences</u> .
s2s+grade	News reports said <u>at least</u> 10 people were <u>held</u> in five Turkish cities.
s2s+grade+PPMI	This year, <u>at least</u> 10 people were <u>held</u> in five Native cities.

を数えた¹¹。

表 8 はエラー分析の例であり、下線は置換すべき箇所および各手法においてそれらに対応する箇所を、太字は省略すべき箇所を示している。例えば、表 8 の上の例では入力文における単語 **stressful** は **tough** に置換すべきだが、いずれのモデルも **hard** へ置換しており、置換失敗としてカウントされる。また、入力文中の句 **immigrant students** はいずれのモデル出力にもそのまま残っており、置換不足としてカウントされる。そして、s2s+grade+PPMI は入力文と正解文に共通して出現する単語 **especially** を **very** へ置換しており、置換過多としてカウントされる。また、下の例では **in five Turkish cities** を省略すべきだが、s2s+grade では省略できていない。

表 9 の分析結果から、省略の成功率（ $= \frac{\text{省略成功}}{\text{省略すべき箇所}}$ ）は s2s+grade が 45 箇所中 13 箇所 (28.9%)、s2s+grade+PPMI が 45 箇所中 19 箇所 (42.2%) であり、また s2s+grade+PPMI は s2s+grade に比べて省略過多は多いが省略不足は少なく、より積極的に省略を行っていることがわかる。

表 10 に置換操作に関するエラー分析結果を示す。本分析では各手法の出力について、省略不足や省略過多ではなかった（省略が成功した）部分のみについて置換すべき箇所をカウントしている。そのため置換すべき箇所の個数が手法によって異なっている。置換の再現率（ $= \frac{\text{置換成功}}{\text{置換すべき箇所}}$ ）は s2s+grade+PPMI が 82 箇所中 4 箇所 (4.9%)、s2s+grade が 86 箇所中 5 箇所 (5.8%) でありいずれも低い。また、置換の適合率（ $= \frac{\text{置換成功}}{\text{置換成功} + \text{置換失敗} + \text{置換過多}}$ ）を見ると、s2s+grade+PPMI が 72 箇所中 4 箇所 (5.6%)、s2s+grade が 74 箇所中 5 箇所 (6.8%) と低く、書き換えの誤りが多い。置換の網羅率（ $= \frac{\text{置換成功} + \text{置換失敗}}{\text{置換すべき箇所}}$ ）は s2s+grade+PPMI が 82 箇所中 38 箇所 (46.3%)、s2s+grade が 86 箇所中 39 箇所 (45.4%) であり、いずれも書き換えようとした箇所自体が半数以下である。

¹¹ 本研究では単一の正解文を用いたが、本来正解文は複数存在するため、正解文とは異なるが正解とすべき出力文が存在する。しかし、ここでは単一正解文の表現のみを正解とする厳しい評価を行うため、本来の性能よりも低く評価される。

表 9 評価用データセットから無作為抽出された 50 文に対する省略のエラー分析

	省略すべき箇所	省略成功	省略不足	省略過多
s2s+grade	45	13	32	16
s2s+grade+PPMI	45	19	26	25

表 10 評価用データセットから無作為抽出された 50 文に対する置換のエラー分析

	置換すべき箇所	置換成功	置換失敗	置換不足	置換過多
s2s+grade	86	5	34	47	35
s2s+grade+PPMI	82	4	34	44	34

表 11 評価用データセットから無作為抽出された 50 文に対する人手評価

	流暢性	意味の等価性	平易性	最小値
s2s+grade	4.36	2.76	3.30	2.40
s2s+grade+PPMI	4.22	2.56	3.54	2.44

5.5.2 人手評価

先行研究 (Zhang and Lapata 2017) に倣い, 流暢性 (Fluency), 意味の等価性 (Adequacy) および平易性 (Simplicity) の 3 項目について人手評価を行った. Zhang and Lapata (2017) と同様に, 各項目について流暢性は「Is the output grammatical and well formed?」, 意味の等価性は「To what extent is the meaning expressed in the original sentence preserved in the output?」, 平易性は「Is the output simpler than the original sentence?」と説明し, 1 から 5 の 5 段階 (1 が最低性能, 5 が最高性能を表す) の評価¹²を依頼した. 評価者は英語話者 1 名である.

表 11 に各項目の平均値を示す. さらに我々は, Mallinson and Lapata (2019) と同様に最小値も評価した. 最小値は, 文ごとに流暢性, 意味の等価性および平易性の各評価値の最小値を取り, 平均したものである. 最小値を用いる理由は, モデルが 3 項目のうちいずれかの項目を無視すれば残りの項目において容易に高いスコアが得られるためである. 例えば, 入力文をそのまま出力すれば流暢性と意味の等価性は最高スコア 5 が得られるが平易性は 1 となる. また意味の等価性を無視すれば平易かつ流暢な文が容易に得られてしまう.

表 11 の最小の列を見ると, s2s+grade+PPMI は, s2s+grade を 0.04 ポイント上回り, s2s+grade と同等以上の性能が確認できた. 流暢性および意味の等価性は s2s+grade の方が, 平易性は s2s+grade+PPMI の方が優れた. 入力から積極的な書き換えを行わない (保守的な) モデルは

¹² 入力文を全く書き換えていない文は平易性が 1 として評価される.

流暢性および意味の等価性が高く、平易性が低くなると考えられるため、s2s+grade+PPMI は s2s+grade と比較してより積極的な書き換えを行っていることがわかる。

5.5.3 今後の課題

エラー分析 (5.5.1 節) の結果、既存手法 s2s+grade と提案手法 s2s+grade+PPMI の両手法とも積極的な省略を行い、置換がほとんど行われなかったことがわかった。その中でも、人手評価 (5.5.2 節) の結果、単語の難易度も考慮した s2s+grade+PPMI は s2s+grade と比較してより積極的な平易化を行うことを確認した。

テキスト平易化では書き換え前後の文で共通した単語が使用される割合が高く、また入力 of 語句の省略も頻繁に行われる。一方、テキスト平易化と同様に系列変換タスクである機械翻訳では入力と出力の言語が異なることから、入力文の語句のほぼ全てが置換され、かつ入力に現れる語句が大幅に省略されることは稀と考えられる。このようなタスクの特性により、テキスト平易化では複製や省略の操作の学習は容易であるが、置換操作は学習が難しく、モデルにとってリスクが高いため積極的な書き換えを行わない。したがって複製や省略という簡単な操作に甘んじやすいと考えられる。そのため今後は、入力文中に出現しない正解単語の出力に大きな報酬を与える工夫が必要であると考えられる。

6 おわりに

本研究では、単語の難易度に注目することで高品質な難易度制御を行うテキスト平易化手法を提案した。素性として単語分散表現に単語の難易度を埋め込む手法、ハードな語彙制約手法、ソフトな語彙制約手法の3種類のアプローチを試みた結果、ソフトな語彙制約手法が有効であった。ソフトな語彙制約では、文の難易度と単語の共起の強さによって損失を重み付けすることで、BLEU および SARI による自動評価が既存手法を上回り、積極的かつ正確な難易度制御に貢献することを確認した。また難易度ごとの分析からも、提案手法が省略および置換の操作に対して有効であることを示した。

本研究では、Scarton and Specia (2018) と同様に、文書の難易度を各文の難易度とみなした。しかし、難解な文書中にも平易な文が含まれる場合があるため、実際には文ごとに難易度を推定する必要がある。テキストの可読性や難易度については、これまで文書 (Kincaid et al. 1975) や語句 (Pavlick and Callison-Burch 2016; Maddela and Xu 2018) の単位では研究されてきたが、文単位での難易度推定は今後の課題である。

謝 辞

本研究は, Microsoft Research Asia, Microsoft Japan Co., Ltd., JST ACT-I Grant Number JPMJPR18UB, JSPS KAKENHI Grant Number JP18K11435 の支援を受けたものである.

参考文献

- Coster, W. and Kauchak, D. (2011). “Simple English Wikipedia: A New Text Simplification Task.” In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp. 665–669.
- Guo, H., Pasunuru, R., and Bansal, M. (2018). “Dynamic Multi-Level Multi-Task Learning for Sentence Simplification.” In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 462–476.
- Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2017). “Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation.” *Transactions of the Association for Computational Linguistics*, **5**, pp. 339–351.
- Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Hermann, U., Aji, A. F., Bogoychev, N., Martins, A. F. T., and Birch, A. (2018). “Marian: Fast Neural Machine Translation in C++.” In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, pp. 116–121.
- Kincaid, J. P., Fishburne Jr., R. P., Rogers, R. L., and Chissom, B. S. (1975). “Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel.” *Technical report, Defence Technical Information Center Document*, pp. 8–75.
- Kingma, D. P. and Ba, J. L. (2015). “Adam: A Method for Stochastic Optimization.” In *Proceedings of the 3rd International Conference on Learning Representations*, pp. 1–15.
- Krashen, S. D. (1985). “The Input Hypothesis: Issues and Implications.” *London: Longman*.
- Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., and Dolan, B. (2016). “A Persona-Based Neural Conversation Model.” In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 994–1003.
- Maddela, M. and Xu, W. (2018). “A Word-Complexity Lexicon and A Neural Readability Ranking Model for Lexical Simplification.” In *Proceedings of the 2018 Conference on Empirical*

Methods in Natural Language Processing, pp. 3749–3760.

Mallinson, J. and Lapata, M. (2019). “Controllable Sentence Simplification: Employing Syntactic and Lexical Constraints.” *arXiv:1910.04387*.

Kajiwar, T. and Komachi, M. (2016). “Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment between Word Embeddings.” In *Proceedings of the 26th International Conference on Computational Linguistics*, pp. 1147–1158.

Nishihara, D., Kajiwar, T., and Arase, Y. (2019). “Controllable Text Simplification with Lexical Constraint Loss.” In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 260–266.

西原大貴, 梶原智之, 荒瀬由紀 (2019). ソフトな語彙制約によるテキスト難易度制御の検討. 情報処理学会第 240 回自然言語処理研究会, **2019-NL-24** (8), pp. 1–6.

Nisioi, S., Štajner, S., Ponzetto, S. P., and Dinu, L. P. (2017). “Exploring Neural Text Simplification Models.” In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 85–91.

Niu, X., Rao, S., and Carpuat, M. (2018). “Multi-Task Neural Models for Translating Between Styles Within and Across Languages.” In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1008–1021.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). “BLEU: A Method for Automatic Evaluation of Machine Translation.” In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318.

Pavlick, E. and Callison-Burch, C. (2016). “Simple PPDB: A Paraphrase Database for Simplification.” In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 143–148.

Pennington, J., Socher, R., and Manning, C. D. (2014). “GloVe: Global Vectors for Word Representation.” In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543.

Petersen, S. E. and Ostendorf, M. (2007). “Text Simplification for Language Learners: A Corpus Analysis.” In *Proceedings of Workshop on Speech and Language Technology in Education*, pp. 69–72.

Scarton, C. and Specia, L. (2018). “Learning Simplifications for Specific Target Audiences.” In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 712–718.

Shardlow, M. (2014). “A Survey of Automated Text Simplification.” *International Journal of Advanced Computer Science and Applications, Special Issue on Natural Language Processing*

2014, pp. 58–70.

- Specia, L. (2010). “Translating from Complex to Simplified Sentences.” In *Proceedings of the 9th International Conference on Computational Processing of the Portuguese Language*, pp. 30–39.
- Sun, H. and Zhou, M. (2012). “Joint Learning of a Dual SMT System for Paraphrase Generation.” In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 38–42.
- Vu, T., Hu, B., Munkhdalai, T., and Yu, H. (2018). “Sentence Simplification with Memory-Augmented Neural Networks.” In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 79–85.
- Wubben, S., van den Bosch, A., and Krahmer, E. (2012). “Sentence Simplification by Monolingual Machine Translation.” In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 1015–1024.
- Xu, W., Callison-Burch, C., and Napoles, C. (2015). “Problems in Current Text Simplification Research: New Data Can Help.” *Transactions of the Association for Computational Linguistics*, **3**, pp. 283–297.
- Xu, W., Napoles, C., Pavlick, E., Chen, Q., and Callison-Burch, C. (2016). “Optimizing Statistical Machine Translation for Text Simplification.” *Transactions of the Association for Computational Linguistics*, **4**, pp. 401–415.
- Zhang, X. and Lapata, M. (2017). “Sentence Simplification with Deep Reinforcement Learning.” In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 584–594.
- Zhao, S., Meng, R., He, D., Andi, S., and Bambang, P. (2018). “Integrating Transformer and Paraphrase Rules for Sentence Simplification.” In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3164–3173.
- Zhu, Z., Bernhard, D., and Gurevych, I. (2010). “A Monolingual Tree-based Translation Model for Sentence Simplification.” In *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 1353–1361.

略歴

西原 大貴：2019 年大阪大学工学部電子情報工学科卒業，同年大阪大学大学院情報科学研究科マルチメディア工学専攻に進学．現在に至る．自然言語処理の研究に従事．

梶原 智之：2013 年長岡技術科学大学工学部電気電子情報工学課程卒業。2015 年同大学大学院工学研究科修士課程電気電子情報工学専攻修了。2018 年首都大学東京大学院システムデザイン研究科博士後期課程情報通信システム学域修了。博士（工学）。同年より大阪大学データビリティフロンティア機構特任助教。自然言語処理の研究に従事。情報処理学会，人工知能学会，言語処理学会各会員。

荒瀬 由紀：2006 年大阪大学工学部電子情報エネルギー工学科卒業。2007 年同大学院情報科学研究科博士前期課程，2010 年同博士後期課程修了。博士（情報科学）。同年，北京の Microsoft Research Asia に入社，自然言語処理に関する研究開発に従事。2014 年より大阪大学大学院情報科学研究科マルチメディア工学専攻准教授，現在に至る。言い換え表現認識と生成，機械翻訳技術，対話システムに興味を持つ。

(2019 年 10 月 1 日 受付)

(2020 年 1 月 15 日 再受付)

(2020 年 2 月 21 日 採録)