

# ソフトな語彙制約によるテキスト難易度制御の検討

西原 大貴<sup>1,a)</sup> 梶原 智之<sup>2,b)</sup> 荒瀬 由紀<sup>1,c)</sup>

**概要:** 本稿では言語学習アプリケーションでの使用を目的として、特定の学年に合わせたテキストの難易度制御を行う。提案手法は、文および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言い換える。まず、文の難易度はテキスト平易化モデルの入力として目標の難易度を加えることで考慮される。また、単語の難易度は目標の難易度に応じて各単語の損失を重み付けすることで考慮される。実験の結果、提案手法が BLEU および SARI の両方を改善することが明らかになった。既存手法は文の難易度のみを考慮しており、文長制御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、提案手法では構文と単語の両方の難易度制御に成功した。

## 1. はじめに

テキスト平易化 [1] 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 [2] 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 [3]。

テキストの難易度は、構文の難易度と単語の難易度によって定まる。表 1 に示す例では、米国の 12 年生（高校 3 年生）向けのテキストでの *According to ~* が 7 年生（中学 1 年生）向けのテキストでは *~ says* に置換されることで、構文的な難しさが減少している。また、単語単位での平易化として、7 年生向けのテキストにおける *Pentagon* が 5 年生（小学 5 年生）向けのテキストでは *military* へ書き換えられている。本研究では、言語学習アプリケーションにおける利用を想定して、このように各学年に応じた難易度へ自動的に書き換えることを目的とする。

表 1 学年に合わせたテキスト平易化の例

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

テキスト平易化における多くの先行研究 [4–11] 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単言語の機械翻訳によって 2 段階の難易度変換を行っており、細かく難易度を制御できない。Scarton and Specia [12] は 11 段階の難易度付きパラレルコーパス [13] を用いて、出力文の目標難易度を入力文にラベルとして付与する手法により、初めて難易度制御を行なった。これにより、同じ入力文であっても目標難易度に応じて言い換え分けることができるようになった。しかし、文全体の難易度のみを考慮しているため、文長など構文の難易度制御には優れるが、ターゲットユーザにとって難易度の高い単語もしばしば出力文に現れる。

そこで本研究では、適切な難易度の単語の出力を促進するために、Scarton and Specia の手法において、各単語の損失を難易度に応じて重み付けする手法を提案する。これにより、既存手法では書き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。評価実験の結果、提案手法は Scarton and Specia の手法と比較して BLEU が 1.04 ポイント、SARI が 0.15 ポイント改善された。さらに詳細な分析の結果、提案手法が単語と構文の両方の難易度を制御し、積極的な書き換えを促進することがわかった。

<sup>1</sup> 大阪大学大学院情報科学研究科

<sup>2</sup> 大阪大学データビリティフロンティア機構

a) nishihara.daiki@ist.osaka-u.ac.jp

b) kajiwara@ids.osaka-u.ac.jp

c) arase@ist.osaka-u.ac.jp

## 2. 関連研究

### 2.1 テキスト平易化

テキスト平易化は難解文から平易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 Wikipedia と Simple Wikipedia のパラレルコーパス (W-SW) [14,15] 上で，2段階の難易度変換モデルを学習してきた．初期の研究では統計的機械翻訳 (SMT) に基づく手法が盛んであったが [4-6]，近年ではニューラル機械翻訳 (NMT) 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が主流である [7-11]．しかし，いずれもテキストの難易度を「難しい」および「易しい」の2段階で変換しており，難易度の制御は行われていない．

### 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 Newsela [13] 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．しかし，Newsela を用いた先行研究の多く [8-11] では，細か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 [12] 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．これは，出力したい目標の難易度を入力文に付与する手法である．具体的には，文の難易度を示す特殊ラベル「<難易度>」を入力文の文頭に付加した上で，Nisioi ら [7] 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．入力文に特殊ラベルを付加して出力文を制御する手法は，多言語翻訳 [16] やスタイル変換 [17] 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 の手法は目標の難易度を考慮しないベースライン [7] と比べて BLEU および SARI 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，文全体の難易度のみを考慮しているため，構文の平易化には優れる一方で，ターゲットユーザにとって難易度の高い単語もしばしば出力文に現れる．そのため本研究では，Scarton and Specia 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. 単語の難易度に基づく損失の重み付け

Scarton and Specia の難易度制御は，文全体の難易度のみを考慮するため，ユーザに適さない難易度の単語もしばしば出力される．そこで本研究では，目標の難易度に適した単語の出力を促すため，図 1 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．本研究では，単語の難易度はその単語の出現する文の難易度と

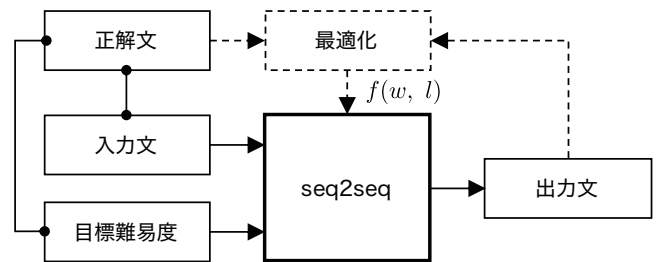


図 1 単語  $w$  と目標難易度  $l$  を用いて損失を重み付けする提案手法

相関を持つと仮定し，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．モデルの出力を  $\mathbf{o} = [o_1, \dots, o_N] \in \mathbb{R}^N$  ( $N$  は語彙数) とすると，損失  $L(\mathbf{o}, \mathbf{y})$  は，

$$L(\mathbf{o}, \mathbf{y}) = -\mathbf{y} \log \mathbf{o}^\top = -\log o_c, \quad (1)$$

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 (1) を

$$L(\mathbf{o}, \mathbf{y}, w, l) = -f(w, l) \log o_c, \quad (2)$$

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．重み  $f(w, l)$  には，TFIDF (Term Frequency Inverse Document Frequency) 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．

$$\text{TFIDF}(w, l) = P(w | l) \log \frac{D}{\text{DF}(w)}. \quad (3)$$

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， $D$  は難易度の総数\*1， $\text{DF}(w)$  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) を考える．

$$\text{PMI}(w, l) = \log \frac{P(w, l)}{P(w)P(l)} = \log \frac{P(w | l)}{P(w)}. \quad (4)$$

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 (3) 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

\*1 本実験では 2 から 12 までの難易度を扱うため  $D = 11$ ．

表 2 実験結果

	BLEU ↑	SARI ↑	BLEU <sub>ST</sub> ↓	文長の MAE ↓	MPMI ↑
source	21.37	2.82	100.0	10.73	0.08
reference	100.0	70.13	18.30	0.00	0.23
s2s	20.43	28.21	37.60	4.38	0.12
s2s+grade	20.82	29.44	31.96	3.77	0.15
s2s+grade+TFIDF	21.00	29.58	31.56	3.75	0.15
s2s+grade+PPMI	<b>21.86</b>	<b>29.59</b>	<b>31.38</b>	<b>3.69</b>	<b>0.19</b>

ち、難易度との PMI が低くなる単語は、難易度に依存せず広く使用される単語であると予測される。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 を用いる。

$$\text{PPMI}(w, l) = \max(\text{PMI}(w, l), 0). \quad (5)$$

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、 $W(w, l)$  はバイアスとして 1 を加えて

$$f(w, l) = \text{Func}(w, l) + 1, \quad (6)$$

$$\text{Func} \in \{\text{PPMI}, \text{TFIDF}\}, \quad (7)$$

とする。

## 4. 評価実験

### 4.1 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に、Scarton and Specia [12] と同じく Newsela を用いた。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より平易なレベルにあるよう書き換えられている。各文書に付与されている難易度は米国の学校制度における学年を表しており、2~12 の 11 段階が用意されている。

本研究で使用した Newsela は、Xu ら [13] によって文アライメントが取られ、Zhang ら [8] 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、先行研究 [12] 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した。

### 4.2 実験設定

本実験では、Marian<sup>\*2</sup> [18] 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

<sup>\*2</sup> <https://github.com/marian-nmt/marian/commit/02f4af4>

は adam 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。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値を報告する。

### 4.3 比較手法

Scarton and Specia [12] は Xu らによって一般に公開されている Newsela よりも大規模なコーパスを利用しており、また訓練・検証・評価用データの分割方法も Zhang ら [8] によるものとは異なっている。そのため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を直接比較することができない。そこで本研究では Scarton and Specia の手法を再現した上で比較を行う。

本実験では、以下の 4 種類の手法を比較した。

- (1) **s2s** は、難易度制御を行わない注意機構に基づく系列変換モデルである。
- (2) **s2s+grade** は、文頭に出力文の目標難易度を付与する既存手法 [12] に基づくモデルである。
- (3) **s2s+grade+TFIDF** は、s2s+grade のベースラインモデルに提案手法 (3 節) である TFIDF に基づく損失の重み付けを適用したモデルである。各単語の TFIDF 値は訓練データを用いて計算した。
- (4) **s2s+grade+PPMI** は、s2s+grade のベースラインモデルに提案手法 (3 節) である PPMI に基づく損失の重み付けを適用したモデルである。各単語の PPMI は訓練データを用いて計算した。

### 4.4 自動評価

本実験では、BLEU [19] および SARI [6] を用いて各手法を自動評価した。これらは、テキスト平易化の研究 [4-12] における標準的な自動評価尺度である。

より詳細な分析のために、本研究では BLEU<sub>ST</sub>、文長の平均絶対誤差 (MAE) および平均 PMI (MPMI) による自動評価も行った。BLEU<sub>ST</sub> は、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、入力文 (Source) と出力文 (Target) の一致度を評価する。これが低いほど、入力文を積極的に書き換えるモデルであると言える。文長の MAE は次式に示す通り、出力文の単語数に基づいて構文の難易度を評価する。

表 3 出力例

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

表 4 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 MPMI

学年	FKGL			MPMI	
	既存手法	提案手法	差分	既存手法	提案手法
<8>	4.92	<b>5.33</b>	+0.41	0.11	<b>0.12</b>
<7>	4.87	<b>5.25</b>	+0.38	0.10	<b>0.12</b>
<6>	4.47	<b>4.56</b>	+0.09	0.12	<b>0.14</b>
<5>	3.51	<b>3.71</b>	+0.20	0.13	<b>0.15</b>
<4>	2.68	<b>2.69</b>	+0.01	0.16	<b>0.19</b>
<3>	<b>2.06</b>	1.89	-0.17	0.18	<b>0.23</b>
<2>	<b>1.81</b>	1.44	-0.37	0.20	<b>0.24</b>
MAE	1.52	<b>1.45</b>	-	-	-

$$MAE_{Len} = \frac{1}{N} \sum_{\substack{s_R \in \text{Reference} \\ s_T \in \text{Target}}} |\text{Len}(s_R) - \text{Len}(s_T)| \quad (8)$$

ここで、 $N$  は評価用データセットの文数、Reference は正解文、 $\text{Len}(s)$  は文  $s$  の単語数を表す。文長の MAE 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。

MPMI は次式に示す通り、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単語の難易度を評価する。

$$MPMI = \frac{1}{W} \sum_{s \in \text{Target}} \sum_{w \in s} \text{PMI}(w, l_s) \quad (9)$$

ここで、 $W$  は出力文 Target に含まれるトークン数、 $l_s$  は文  $s$  の目標難易度である。PMI は訓練データを用いて式 (4) によって求めた。MPMI 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。

## 5. 実験結果と分析

### 5.1 実験結果

表 2 に示した評価結果のうち、source は入力文を変え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ものである。表 2 によると、提案手法は、BLEU と SARI の両方において既存手法である s2s+grade を上回った。特に s2s+grade+PPMI の BLEU と SARI は s2s+grade と比較してそれぞれ 1.04 ポイント、0.15 ポイント改善した。

BLEU<sub>ST</sub> は、提案手法が最も優れていた。特に、s2s+grade+PPMI が最も積極的に書き換えを行なって

いる。したがって、提案手法は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より積極的に出力すると分かる。s2s は目標の難易度を考慮しないため、積極的な書き換えを学習せずあまり書き換えを行わない保守的なモデルとなり、BLEU<sub>ST</sub> が他の手法と比較して高くなったと考えられる。

文長の MAE を見ると、s2s+grade は s2s から 0.61 ポイント改善され、2 つの提案手法は 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、構文の難易度制御が改善されている。MPMI の観点では、s2s+grade+PPMI が最も高い。したがって s2s+grade+PPMI は、s2s+grade による構文の難易度制御の特性を保ちつつ、適した難易度の単語を出力できることが明らかになった。

表 3 にモデルの出力例を示す。s2s+grade+PPMI は incarnation を 7 年生向けとして people に、motivate を 4 年生向けとして inspire に書き換えている。いずれのモデルも and , on ~ を削除できているが、s2s+grade+PPMI は 4 年生向けの文をより短い文を出力できている。

### 5.2 各難易度における FKGL と MPMI による分析

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\*3 に向けての平易化を行った。表 4 に、s2s+grade (既存手法) と s2s+grade+PPMI (提案手法) に対する目標難易度ごとの FKGL [20] および MPMI を示す。FKGL は平均音節数および平均文長に基づきテキストの可読性を評価する自動評価尺度である。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、本実験ではこれを目標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、目標学年に対する MAE を比較すると提案手法は既存手法より優れている。表 4 中、FKGL の差分は学年ごとに既存手法と提案手法の FKGL の差を示している。これによると、提案手法は目標学年が

\*3 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。

表 5 エラー分析の例

入力文	<u>She said the college application process can be especially stressful for immigrant students .</u>
正解文	Albrecht said applying for college can be especially tough for immigrants .
既存手法	She said the college program can be especially hard for immigrant students .
提案手法	She said it can be very hard for immigrant students .

表 6 評価用データセットから無作為抽出された 25 文に関するエラー分析

	省略不足	省略過多	置換すべき箇所	置換成功	置換失敗	置換不足	置換過多
既存手法	14	9	50	2	20	28	21
提案手法	11	15	48	4	23	21	19

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、既存手法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。この結果により、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。

M PMI の評価によると、提案手法は全ての学年で既存手法より優れている。したがって、PPMI 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。

### 5.3 エラー分析

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 25 文について、s2s+grade (既存手法) および s2s+grade+PPMI (提案手法) のエラー分析を行った。ここでは、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置換について、それぞれの細分類ごとにエラー数を人手で調べた。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、省略すべきだが省略しなかった箇所 (省略不足) と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) を数えた。<sup>\*4</sup>

表 5 はエラー分析の例であり、下線は編集箇所を示している。この例では、省略の操作は行われていないことに注意されたい。例えば、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、提案手法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。

表 6 の分析結果から、提案手法は既存手法に比べて、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。置換の再現率 ( $= \frac{\text{置換成功}}{\text{置換すべき箇所}}$ ) は提案手法が  $\frac{4}{48}$ 、既存手法が  $\frac{2}{50}$  であり、提案手法がより正しく置換できたものの、いずれも低い。また、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、提案手法が  $\frac{2}{43}$ 、既存手法が  $\frac{4}{46}$  と低く、書き換えの誤りが多い。一方、網羅率 ( $= \frac{\text{置換失敗} + \text{置換成功}}{\text{置換すべき箇所}}$ ) は提案手法が  $\frac{27}{48}$ 、既存手法が  $\frac{22}{50}$  である。テキスト平易化においては、難解な表現 (置換すべき箇所) を平易に言い換える (置換成功) ことが目的であるため、高い置換の再現率を達成することが望ましい。既存手法と提案手法はいずれも、置換の網羅率に比べて適合率が低いために再現率が低いことがわかる。単語の難易度も考慮してテキストの難易度制御を行う提案手法は、より積極的な省略および置換を行うが、より正しい単語を出力することが今後の課題である。

## 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。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ができることを確認した。また難易度ごとの分析からも、提案手法が構文的にも語彙的にもより正確な難易度制御を実現できることを確認した。

本研究では、Scarton and Specia [12] と同様に、文書の難易度を各文の難易度とみなした。テキストの可読性や難易度については、文書 [20] や語句 [21, 22] の単位では研究されているが、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 の助成を受けたものです。

<sup>\*4</sup> 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、正解文と異なる場合であっても正しい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なうため、本来の評価よりも低く評価される。

## 参考文献

- [1] 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, p. 425 (2014).
- [2] Krashen, S. D.: The input hypothesis: Issues and implications, *New York: Longman* (1985).
- [3] Petersen, S. E. and Ostendorf, M.: Text Simplification for Language Learners: A Corpus Analysis, *Speech in Language Technology in Education (SLaTE2007)*, No. SLaTE, p. 4 (2007).
- [4] Specia, L.: Translating from Complex to Simplified Sentences, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 6001 LNAI, pp. 30–39 (2010).
- [5] Wubben, S., van den Bosch, A. and Krahrmer, E.: Sentence Simplification by Monolingual Machine Translation, *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Vol. 1, pp. 1015–1024 (2012).
- [6] Xu, W., Napoles, C., Pavlick, E., Chen, Q. and Callison-Burch, C.: Optimizing Statistical Machine Translation for Text Simplification, *Transactions of the Association for Computational Linguistics*, Vol. 4, No. 1, pp. 401–415 (2016).
- [7] Nisioi, S., Štajner, S., Ponzetto, S. P. and Dinu, L. P.: Exploring Neural Text Simplification Models, *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vol. 2, pp. 85–91 (2017).
- [8] Zhang, X. and Lapata, M.: Sentence Simplification with Deep Reinforcement Learning, *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 584–594 (2017).
- [9] Vu, T., Hu, B., Munkhdalai, T. and Yu, H.: 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*, Vol. 2, pp. 79–85 (2018).
- [10] Guo, H., Pasunuru, R. and Bansal, M.: Dynamic Multi-Level Multi-Task Learning for Sentence Simplification, *In Proceedings of the 27th International Conference on Computational Linguistics*, pp. 462–476 (2018).
- [11] Zhao, S., Meng, R., He, D., Andi, S. and Bambang, P.: Integrating Transformer and Paraphrase Rules, *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3164–3173 (2018).
- [12] Scarton, C. and Specia, L.: Learning Simplifications for Specific Target Audiences, *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 712–718 (2018).
- [13] Xu, W., Callison-Burch, C. and Napoles, C.: Problems in Current Text Simplification Research: New Data Can Help, *Transactions of the Association for Computational Linguistics*, Vol. 3, pp. 283–297 (2015).
- [14] Zhu, Z., Bernhard, D. and Gurevych, I.: A Monolingual Tree-based Translation Model for Sentence Simplification, *Computational Linguistics*, No. August, pp. 1353–1361 (2010).
- [15] Coster, W. and Kauchak, D.: Simple English Wikipedia: A New Text Simplification Task William, *Research Journal of Agricultural Sciences*, pp. 665–669 (2011).
- [16] 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.: Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, *Transactions of the Association for Computational Linguistics*, pp. 339–351 (2016).
- [17] Niu, X., Rao, S. and Carpuat, M.: 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 (2018).
- [18] Junczys-Dowmunt, M., Heafield, K., Hoang, H., Grundkiewicz, R. and Aue, A.: Marian: Cost-effective High-Quality Neural Machine Translation in C++, *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, pp. 116–121 (2018).
- [19] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: 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 (2002).
- [20] Kincaid, J. P., Fishburne Jr., R. P., Rogers, R. L. and Chissom, B. S.: 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* (1975).
- [21] Pavlick, E. and Callison-Burch, C.: Simple PPDB: A Paraphrase Database for Simplification, *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 143–148 (2016).
- [22] Maddela, M. and Xu, W.: 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 (2018).