E-032

やさしい日本語へのテキスト平易化のための訓練データの精選

Towards Parallel Corpus Filtering for Japanese Text Simplification

畠垣 光希†，　梶原 智之†，　二宮 崇†
Koki Hatagaki,　Tomoyuki Kajiwara,　Takashi Ninomiya

1. はじめに

留学生30万人計画や技能実習制度などの政府の方針およびグローバル化の進行により，日本における在留外国人数が増加しており，その数は 300 万人[1]に到達しようとしている．2010 年の調査 [1] では，在留外国人のうち日本語を理解できる人数は 62%と，英語の 44%や中国語の 38%を大きく上回り，在留日本人にとって最も広く理解される言語が日本語であると報告されている．このような背景を受け，災害情報の速報[2]や日々のニュース発信[3]など，様々な場面で「やさしい日本語」による情報提供が進められている．

自然言語処理の分野では，機械学習の技術を用いて所与の日本語文をやさしい日本語に自動的に言い換える，日本語を対象としたテキスト平易化の研究 [2-7] が行われている．テキスト平易化とは，「署名してください」という文を「名前を書いてください」と言い換えるように，ある文の表現をより平易な同義表現に変換するタスクである．

日本語におけるテキスト平易化の先行研究 [4] では，自動生成された平易文の文法的な正しさや難解文と平易文の間の意味的類似性に関する人手評価は高いものの，平易文の易しさに関しては不十分であるという課題が報告されている．その原因のひとつとして，訓練に用いたやさしい日本語コーパス [5-7] に無理な平易化をしている文対が存在することが考えられる．実際，やさしい日本語コーパスには，「ピザは私の大好物です」という難解文に対して「丸く伸ばしたパイにチーズなどを置いて焼いたものは私のとても好きな食べ物です」という平易文を付与した不自然な文対が含まれる．このような文対がノイズとなり，テキスト平易化モデルに悪影響を与えている可能性がある．

そこで本研究では，図 1 に示すように，上述したようなノイズとなる文対をやさしい日本語コーパスの訓練データから除去するパラレルコーパスフィルタリングに取り組み，日本語のテキスト平易化モデルの性能を改善する．3.1 節で説明するように，やさしい日本語コーパスには主に以下の 3 種類のノイズが含まれる．

- 文長差が大きい文対
- 同義性が低い文対
- 流暢性が低い文を含む文対

本研究では，それぞれのノイズ文対を検出するための手法を提案し，パラレルコーパスフィルタリングによるテキスト平易化の性能の変化を評価する．文長差に関する手法では，難解文と平易文の文字・単語・サブワード単位の文長差や編集距離を用いたパラレルコーパスフィルタリングを行う．同義性に関する手法では，単語や文の分散表現を用いたパラレルコーパスフィルタリングを行う．流暢性に関

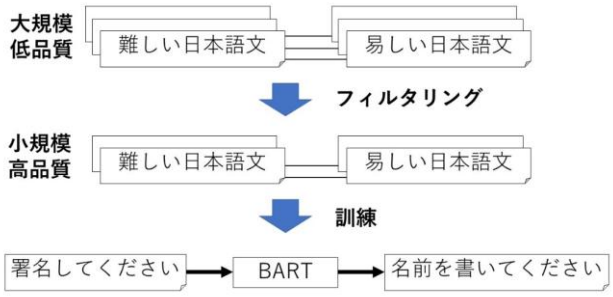

図 1：本研究の流れ

する手法では，言語モデルによる文の尤度を用いたパラレルコーパスフィルタリングを行う．テキスト平易化モデルは，大規模コーパスを用いて事前訓練された系列変換モデル BART [8] を利用し，やさしい日本語コーパスを用いてファインチューニングする．

評価実験の結果，特に文長差に関するパラレルコーパスフィルタリングが有効であることが明らかになった．テキスト平易化タスクにおける自動評価指標である BLEU および SARI の両方で，やさしい日本語コーパスの全体を用いて訓練するベースラインを上回る性能を達成した．

2. 関連研究

本節では，テキスト平易化の関連研究およびパラレルコーパスフィルタリングの基盤技術について概説する．

2.1 テキスト平易化のモデル

難解な文を平易に言い換えるテキスト平易化 [9] は，同一言語内の機械翻訳タスクと考えられる．そのため，難解な文と平易な文からなるパラレルコーパス [10] を用いて系列変換モデルを訓練するアプローチが主流である．2010 年から統計的機械翻訳モデルに基づくテキスト平易化 [10-12] が研究されており，2010 年代の後半からはニューラル機械翻訳モデルに基づくテキスト平易化 [13-15] が研究されている．現在では，自己注意機構に基づく Transformer [16] やそれを大規模コーパス上で事前訓練した BART [8] が，テキスト平易化において高い性能 [17-19] を達成している．

Transformer [16] は，逐次的に単語系列を処理する再帰型ニューラルネットワークの計算効率を改善しつつ，自己注

†愛媛大学　Ehime University
[1] https://www.moj.go.jp/isa/content/001356650.pdf
[2] https://twitter.com/sendai_kiki2
[3] https://www3.nhk.or.jp/news/easy/

意機構によって単語間の関係性を考慮できる系列変換モデルである．現在では，機械翻訳をはじめとして様々なテキスト生成タスクにおいて利用されている．近年の自然言語処理では，大規模コーパス上での単語穴埋めの事前訓練を経た Transformer を目的タスクにおいてファインチューニングする転移学習のアプローチが多くの応用タスクにおいて広く採用されている．感情分析などの分類タスクで用いられる BERT [20] のマスク言語モデリングと同様に，テキスト平易化などの生成タスクで用いられる BART [8] のノイズ除去自己符号化の事前訓練では，入力文の一部のトークンをマスクし，そのトークンを復元する．BERT と BART の違いは，前者がエンコーダのみを用いてマスクトークンを復元する一方で，後者はエンコーダ・デコーダモデルを用いてマスクトークンを復元することである．

BART では事前訓練の恩恵を受けて，目的タスクにおける小規模なパラレルコーパス上でのファインチューニングから高品質なテキスト生成モデルを得ることができる．テキスト平易化の先行研究は，パラレルコーパスマイニング [21-23] やデータ拡張 [24] による訓練データの大規模化に焦点を当ててきた．しかし，BART の登場によって，訓練データの規模だけでなく品質についても再考すべき時期が到来しており，本研究では訓練データからノイズを除去するパラレルコーパスフィルタリングに取り組む．

2.2 テキスト平易化のパラレルコーパスフィルタリング

パラレルコーパスフィルタリング [25] は，大規模な訓練用パラレルコーパスを入手可能な機械翻訳タスクにおいて主に研究されている技術であり，ノイズを含む文対を訓練データから除去することによって生成文の品質改善に貢献している．テキスト平易化タスクは比較的小規模な訓練データを用いるため，パラレルコーパスフィルタリングの先行研究は存在しない．ただし，文単位の対応関係が明らかではない記事の対（コンパラブルコーパス）から対応する難解文と平易文の対を自動抽出するパラレルコーパスマイニング [21-23] は盛んに研究されており，これらの研究の中でテキスト平易化タスクにおける文対の対応関係の良さを推定する技術が検討されている．また，テキスト平易化モデルにおける出力文のリランキング [26] や強化学習の際の報酬計算 [14] の中でも，難解文と平易文の対応の良さを推定する技術が考えられている．特に，文対の同義性を推定する技術や文の流暢性を推定する技術は，本研究のパラレルコーパスフィルタリングと密接に関連する．

同義性については，fastText などの単語分散表現 [27] や Universal Sentence Encoder などの文分散表現 [28] に基づく文間の意味的類似度推定 [22,26] が行われている．単語分散表現に基づく手法では，文中に出現する各単語の分散表現の平均プーリング [29] の間の余弦類似度や単語分散表現のアライメント [30] によって，文対の同義性を推定できる．文分散表現に基づく手法では，分散表現の間の余弦類似度によって，文対の同義性を推定できる．

文法性については，GPT-2 などの単方向言語モデル [31] や BERT などの双方向言語モデル [20] に基づく文の尤度推定 [14,26] が行われている．単方向言語モデルに基づく手法では，$i-1$番目までのトークンからi番目のトークンを推定する際の対数確率によって，文の流暢性を推定できる．

表 1：やさしい日本語コーパスに含まれるノイズの内訳

ノイズの種類	割合（重複あり）
文長差が大きい文対	4 %（ 20/500 件）
同義性が低い文対	8 %（ 42/500 件）
流暢性が低い文を含む文対	8 %（ 41/500 件）
その他のノイズを含む文対	1 %（ 2/500 件）
ノイズを含まない文対	84 %（419/500 件）

先行研究で用いられている N-gram 言語モデルでは直前の$N-1$トークンのみを文脈として考慮するが，GPT-2 のようなニューラル言語モデルでは全ての文内文脈を考慮できる．双方向言語モデルに基づく手法では，文中のトークンをひとつずつ順番にマスクした際のマスクトークンの対数確率 [32] によって，文の流暢性を推定できる．

3. 提案手法

本研究では，日本語のテキスト平易化の性能向上を目的として，やさしい日本語コーパス [5-7] に含まれるノイズを対象に，パラレルコーパスフィルタリングの手法を提案する．訓練データに対して複数の手法でパラレルコーパスフィルタリングを実施し，得られた訓練データのサブセットを用いてテキスト平易化モデルを訓練する．モデルには事前訓練された BART [8] を使用し，やさしい日本語コーパスのサブセットを用いたファインチューニングによってテキスト平易化モデルを構築する．

まず 3.1 節では，やさしい日本語コーパスに含まれるノイズについて分析し，本研究で扱う代表的な 3 種類のノイズについて定義する．そして 3.2 節から 3.4 節では，それぞれのノイズ文対を検出するための提案手法を説明する．

3.1 やさしい日本語コーパスに含まれるノイズの定義

やさしい日本語コーパスから無作為抽出した 500 文対について，含まれるノイズを第 1 著者が人手で分類した結果を表 1 に示す．やさしい日本語コーパスは，所与の文に対して平易な言い換え文を付与したものである．人手で言い換えを付与しているため，大部分の文対はノイズを含まないと期待できる．

我々の分析からも，84%の文対がノイズを含まないことを確認できた．ノイズを含む文対（500 文対のうち 81 文対）を分類したところ，文長差が大きい文対・同義性が低い文対・流暢性が低い文対の 3 種類が主要なノイズであることがわかった．これらの例を表 2 に示す．文長差が大きい文対には，難解な語句を辞書の定義文のような表現で置換したような例が多く見られた．同義性が低い文対には，書き換えの際に関連するが同じ意味ではない表現を使用してしまった失敗例が見られた．流暢性が低い文には，助詞の誤りなどの単純なミスとともに，無理な平易化による冗長な表現が多く見られた．

3.2 文長差に関する手法

本手法では，難解文と平易文の間の文長差が大きいノイズを検出し，パラレルコーパスフィルタリングを行う．や

表 2：やさしい日本語コーパスに含まれるノイズの例

ノイズの種類	難解文	平易文
文長差が大きい文対	それの代金を仕払うことによって確立する所有権	買う
	このひもは強い	この物を制限するための長いものは強い
	洪水がおさまり始めた	水の量が増えて川から出る状態が静かになり始めた
同義性が低い文対	くじで誰が勝つか決めよう	勝ったか負けたか決めることができない
	熱はたいていの物を膨張させる	あらゆる物は熱で増える
	彼女はみんなをうんざりさせます	彼女はみんなを飽きさせます
流暢性が低い文を含む文対	豆腐は良い酒の肴になる	植物で作った白い柔らかい物を食べると，うまい酒がたくさん飲むことができる
	金融引き締めで金利が上昇するだろう	金の流れを厳しくすることで金を借りる際に返す時につける金が占める率のが上がるだろう
	酸が金属を腐食した	酸っぱい特徴を持つ水が金属を腐らせた

さしい日本語コーパスに含まれる文対のうち，本手法は表 2 の上段に示す種類のノイズを扱う．これらのノイズには，1 件目のように極端な平易化によって過度に情報を落としてしまった例や，2 件目や 3 件目のように辞書の定義文のような表現で難解な語句を置換した例が含まれる．

文対の文長差を求めるために，本研究ではトークン数の差の絶対値を用いる手法とトークン単位の編集距離を用いる手法の 2 つを提案する．トークンには，文字・単語・サブワードの 3 種類を採用する．閾値よりも文長差が大きい文対をノイズとして検出し，訓練データから除去する．

3.3 同義性に関する手法

本手法では，難解文と平易文の文間の意味的類似度が小さいノイズを検出し，パラレルコーパスフィルタリングを行う．やさしい日本語コーパスに含まれる文対のうち，本手法は表 2 の中段に示す種類のノイズを扱う．これらのノイズには，「たいていの物」と「あらゆる物」や「うんざりさせる」と「飽きさせる」のように，関連するが同じ意味ではない表現を用いた書き換えが含まれる．

文間の意味的類似度を推定するために，本研究では単語分散表現および文分散表現に基づく合計 3 つの手法を提案する．単語分散表現に基づく手法では，fastText [27] の日本語モデルを用いる．文分散表現に基づく手法では，Universal Sentence Encoder [28] の多言語版である mUSE [33] を用いる．文間の意味的類似度が閾値よりも低い文対をノイズとして検出し，訓練データから除去する．

3.3.1 単語分散表現の平均プーリングによる同義性の推定

単語分散表現に基づいて文間の意味的類似度を推定するために，単語分散表現の平均プーリングによって文分散表現を構成する手法 [29] を用いる．この手法は先行研究 [2,22] においてもベースライン手法として採用されており，このようにして得た文分散表現の間の余弦類似度によって，以下のように文間の意味的類似度を推定する．ただし，x は難解文，yは平易文であり，$|x|$および$|y|$は各文の単語数，x_iは文x中のi番目の単語，$\vec{x}_i$はその単語分散表現を表す．

$$\mathrm{AES}(x,y)=\cos\left(\frac{1}{|x|}\sum_{i=1}^{|x|}\vec{x}_i,\frac{1}{|y|}\sum_{j=1}^{|y|}\vec{y}_j\right)$$

3.3.2 単語分散表現のアライメントによる同義性の推定

単語分散表現に基づいて文間の意味的類似度を推定するために，先行研究 [2,22] でも用いられている単語分散表現のアライメント手法 [30] を用いる．この手法では，文間の単語アライメントの問題を，単語分散表現をノード，単語分散表現間の余弦類似度をエッジの重みとする重み付き完全 2 部グラフのマッチング問題として考え，最大マッチングによって単語アライメントを得る．そして，対応付けられた単語間の単語分散表現の余弦類似度を平均して文間の意味的類似度を推定する．

$$\mathrm{MAS}(x,y)=\frac{1}{2|x|}\sum_{i=1}^{|x|}\max_j\cos(\vec{x}_i,\vec{y}_j)+\frac{1}{2|y|}\sum_{j=1}^{|y|}\max_i\cos(\vec{x}_i,\vec{y}_j)$$

3.3.3 文分散表現による同義性の推定

文間の意味的類似度を推定するために，mUSE による文分散表現の余弦類似度を用いる．この手法では，単語分散表現に基づく手法では考慮されなかった文内の周辺単語の文脈情報を活用できる．BERT [20] などの近年の汎用的な文符号化器は，ファインチューニングなしで文間の意味的類似度を適切に推定することは難しい．日本語では，文間の意味的類似度推定のために使用可能なラベル付きコーパスを利用できないため，本研究ではファインチューニングなしで文間の意味的類似度推定が可能な mUSE を用いる．

3.4 流暢性に関する手法

本手法では，平易文の流暢性が低いノイズを検出し，パラレルコーパスフィルタリングを行う．やさしい日本語コーパスに含まれる文のうち，本手法は表 2 の下段に示すノイズを扱う．これらのノイズには，1 件目や 2 件目のように助詞などを誤った例や，3 件目のように不自然な言い回しをしている例が含まれる．

文の流暢性を推定するために，本研究では言語モデルに基づく 2 つの手法を提案する．単方向言語モデルに基づく手法では，GPT-2 [31] の日本語モデルを用いる．双方向言語モデルに基づく手法では，BERT [20] の日本語モデルを用いる．閾値よりも高いパープレキシティを持つ文を含む文対をノイズとして検出し，訓練データから除去する．

3.4.1 単方向言語モデルによる流暢性の推定

文の流暢性を推定するために，単方向言語モデルに基づくパープレキシティを用いる．先行研究 [14,26] では N-gram 言語モデルが用いられているが，本研究では GPT-2 のニューラル言語モデルを用いることで，全ての文内文脈を考慮する．パープレキシティは以下のように計算する．ただし，$P(x_i|x_{<i})$はi番目より前の単語が与えられた際にi番目の単語を出力する条件付き確率である．

$$\mathrm{PPL}(x) = \exp\left(-\frac{1}{|x|}\sum_{i=1}^{|x|}\log P(x_i|x_{<i})\right)$$

3.4.2 双方向言語モデルによる流暢性の推定

文の流暢性を推定するために，双方向言語モデルに基づく擬似的なパープレキシティ [32] を用いる．単方向言語モデルに基づくパープレキシティが過去の単語系列から次の単語を推定する際の条件付き確率の対数尤度の総和である一方で，双方向言語モデルに基づくパープレキシティは周辺単語からマスクされた単語を推定する際の条件付き確率の対数尤度の総和であり，以下のように計算する．ただし，$x_{\setminus i}$は文xに含まれる単語のうちi番目以外の単語たちを表す．

$$\mathrm{pseudoPPL}(x) = \exp\left(-\frac{1}{|x|}\sum_{i=1}^{|x|}\log P(x_i|x_{\setminus i})\right)$$

4. 評価実験

日本語のテキスト平易化タスクにおける評価実験によって，提案手法によるパラレルコーパスフィルタリングの有効性を検証する．本節の構成を説明する．まず 4.1 節ではデータセットと評価指標について，次に 4.2 節ではモデルやハイパーパラメータなどの実験設定について，そして 4.3 節では検証データを用いた閾値の設定について述べる．4.4 節では評価データにおける実験結果を示し，最後に 4.5 節では実験結果に対する詳細な分析を行う．

4.1 データセットおよび評価指標

本研究では，日本語のテキスト平易化のためのパラレルコーパスであるやさしい日本語コーパス[4,5] [5-7] を用いて，提案手法によるパラレルコーパスフィルタリングの有効性を検証した．やさしい日本語コーパスは，人手で作成された 85,000 件の難解文と平易文の文対である．このうち 5 万文対は日本語母語話者の大学生によってアノテーションされており，3.5 万文対はクラウドソーシングによって雇用された日本語母語話者によってアノテーションされている．本実験では，難解文に対して 7 種類の平易文が付与されたマルチリファレンスの 100 文対を評価データ，その他のシングルリファレンス部分から無作為抽出された 2,000 文対を検証データとして，残りの 82,300 文対をパラレルコーパスフィルタリングの対象となる訓練データとして使用した．

テキスト平易化モデルの性能評価には，本タスクで一般的に用いられる BLEU [34] および SARI [35] の自動評価指標を用いた．BLEU は出力文と正解文の間の表層的な一致度を評価し，SARI は入力文・出力文・正解文の比較から編集操作の良さを評価する．BLEU および SARI の実装には，EASSE[6] [36] を用いた．なお，EASSE では自動評価の前処理として，Mecab[7] [37] による単語分割が行われている．

訓練データ全体を用いて訓練したテキスト平易化モデルの性能と，提案手法によって抽出された訓練データのサブセットを用いて訓練したテキスト平易化モデルの性能を，BLEU および SARI でそれぞれ評価して比較することによって，パラレルコーパスフィルタリングの有効性を検証する．

4.2 実験設定

テキスト平易化モデルには，日本語 Wikipedia を用いて事前訓練された日本語 BART[8] [8] を用いた．fairseq[9] [38] を用いて実装し，やさしい日本語コーパスにおけるファインチューニングの際には，最適化手法として Adam [39] を使用し，$\beta_1 = 0.9$，$\beta_2 = 0.99$ とし，学習率は 5e-4 と設定した．バッチサイズは 4,096 トークンとし，正則化にはラベル平滑化および dropout を用いた．なお，dropout 確率は 0.2 とした．また，検証データのクロスエントロピー損失に基づき，5 回連続で改善が見られない場合に訓練を終了する early-stopping を採用した．

前処理として，テキスト平易化モデルに入力する難解文には Juman++[10] [40] による単語分割に加えて SentencePiece[11] [41] によるサブワード分割を行った．なお，サブワード分割の際の語彙サイズは 8,000 に設定した．

提案手法のパラレルコーパスフィルタリングのために，以下のモデルを使用した．単語分散表現には，fastText [27] の日本語モデル[12]を使用した．このときの単語分割には，MeCab [37] を用いた．文分散表現には，Universal Sentence Encoder [28] の多言語版である mUSE[13] [33] を使用した．単方向言語モデルには，GPT-2 [31] の日本語モデル[14]を使用した．双方向言語モデルには，BERT [20] の日本語モデル[15]を使用した．GPT-2 および BERT の言語モデルは，HuggingFace Transformers[16] [42] を用いて実装した．

4.3 検証データを用いた閾値の設定

本節では，検証データにおけるテキスト平易化の性能を評価し，各手法におけるパラレルコーパスフィルタリングの閾値を設定する．本研究では，テキスト平易化のための主要な自動評価指標である SARI を基準に，検証データにおける最高の性能を達成する閾値を採用した．

[4] https://www.jnlp.org/GengoHouse/snow/t15
[5] https://www.jnlp.org/GengoHouse/snow/t23
[6] https://github.com/feralvam/easse
[7] http://taku910.github.io/mecab/
[8] https://nlp.ist.i.kyoto-u.ac.jp/?BART 日本語 Pretrained モデル
[9] https://github.com/pytorch/fairseq
[10] https://github.com/ku-nlp/jumanpp
[11] https://github.com/google/sentencepiece
[12] https://fasttext.cc/docs/en/crawl-vectors.html
[13] https://tfhub.dev/google/universal-sentence-encoder-multilingual/3
[14] https://huggingface.co/rinna/japanese-gpt2-medium
[15] https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking
[16] https://github.com/huggingface/transformers

表 3：検証データにおける文長差に関するパラレルコーパスフィルタリングの性能評価

手法	閾値	削除した文対数	BLEU	SARI
ベースライン		0	71.69	61.06
トークン数の差（文字）	8	3,972	71.38	60.03
トークン数の差（文字）	9	3,106	71.03	60.61
トークン数の差（文字）	10	2,497	71.57	**61.36**
トークン数の差（文字）	11	1,988	71.59	60.72
トークン数の差（文字）	12	1,583	71.62	60.70
トークン数の差（単語）	11	6,111	70.84	60.58
トークン数の差（単語）	12	5,173	70.96	60.12
トークン数の差（単語）	13	4,450	71.65	**61.37**
トークン数の差（単語）	14	3,834	71.40	60.21
トークン数の差（単語）	15	3,302	71.65	60.72
トークン数の差（サブワード）	3	5,813	71.63	60.83
トークン数の差（サブワード）	4	3,142	71.45	60.63
トークン数の差（サブワード）	5	1,792	71.36	60.75
トークン数の差（サブワード）	6	1,116	71.26	**61.11**
トークン数の差（サブワード）	7	725	71.30	60.30
編集距離（文字）	13	6,305	71.48	60.44
編集距離（文字）	14	5,446	71.39	60.38
編集距離（文字）	15	4,538	71.63	**61.52**
編集距離（文字）	16	3,830	**71.82**	60.40
編集距離（文字）	17	3,235	71.63	61.01
編集距離（単語）	8	6,966	**71.72**	60.83
編集距離（単語）	9	5,297	71.49	**61.58**
編集距離（単語）	10	4,065	71.53	60.67
編集距離（単語）	11	3,099	71.14	60.05
編集距離（単語）	12	2,382	71.67	60.37
編集距離（サブワード）	6	11,555	71.50	60.57
編集距離（サブワード）	7	8,374	70.87	59.73
編集距離（サブワード）	8	6,004	70.95	60.73
編集距離（サブワード）	9	4,400	71.60	60.60
編集距離（サブワード）	10	3,202	71.28	60.34

4.3.1 文長差に関する手法の閾値

文長差に関する手法について，閾値を変更しながらパラレルコーパスフィルタリングを実施し，検証データにおけるテキスト平易化の性能を評価した結果を表 3 に示す．

文字数の差に基づく手法では，トークン数の差が 10 を超える文対を除去することで検証データにおける最高の SARI が得られた．このとき除去された文対数は 2,497 件（訓練データの 3.03%）であった．また，単語数の差に基づく手法では，トークン数の差が 13 を超える文対を除去することで検証データにおける最高の SARI が得られた．このとき除去された文対数は 4,450 件（訓練データの 5.40%）であった．そして，サブワード数の差に基づく手法では，トークン数の差が 6 を超える文対を除去することで検証データにおける最高の SARI が得られた．このとき除去された文対数は 1,116 件（訓練データの 1.36%）であった．

文字単位の編集距離に基づく手法では，編集距離が 15 を超える文対を除去することで検証データにおける最高の SARI が得られた．このとき除去された文対数は 4,538 件（訓練データの 5.51%）であった．また，単語単位の編集距離に基づく手法では，編集距離が 9 を超える文対を除去することで検証データにおける最高の SARI が得られた．このとき除去された文対数は 5,297 件（訓練データの 6.43%）であった．なお，サブワード単位の編集距離に基づく手法では，編集距離が 8 を超える文対を除去することで検証データにおける最高の SARI が得られたものの，訓練データの全体を用いて訓練するベースラインの性能を上回ることはなかった．

4.3.2 同義性に関する手法の閾値

同義性に関する手法について，閾値を変更しながらパラレルコーパスフィルタリングを実施し，検証データにおけるテキスト平易化の性能を評価した結果を表 4 に示す．

平均単語分散表現に基づく手法では，類似度が 0.9 を下回る文対を除去することで検証データにおける最高の SARI が得られた．このとき除去された文対数は 5,708 件（訓練データの 6.94%）であった．なお，単語アライメントに基づく手法では，類似度が 0.7 を下回る文対を除去することで検証データにおける最高の SARI が得られたものの，訓練データの全体を用いて訓練するベースラインの性能を上回ることはなかった．

文分散表現に基づく手法では，類似度が 0.5 を下回る文対を除去することで検証データにおける最高の SARI を得

表 4：検証データにおける同義性に関するパラレルコーパスフィルタリングの性能評価

手法	閾値	削除した文対数	BLEU	SARI
ベースライン		0	71.69	61.06
平均単語分散表現	0.85	1,553	71.51	60.33
平均単語分散表現	0.90	5,708	**71.71**	60.40
平均単語分散表現	0.95	20,525	70.94	59.23
単語アライメント	0.4	233	71.62	60.47
単語アライメント	0.5	1,353	71.52	60.78
単語アライメント	0.6	4,819	71.44	60.44
単語アライメント	0.7	12,533	71.54	60.87
単語アライメント	0.8	26,281	70.50	57.93
文分散表現	0.4	832	71.64	60.66
文分散表現	0.5	2,312	**71.71**	60.87
文分散表現	0.6	5,519	71.54	60.06
文分散表現	0.7	11,547	71.10	60.83
文分散表現	0.8	21,842	70.78	58.48

表 5：検証データにおける流暢性に関するパラレルコーパスフィルタリングの性能評価

手法	閾値	削除した文対数	BLEU	SARI
ベースライン		0	71.69	61.06
単方向言語モデル	40	3,053	71.63	60.58
単方向言語モデル	60	894	70.95	60.73
単方向言語モデル	80	399	71.44	60.69
単方向言語モデル	100	227	71.55	60.83
双方向言語モデル	150	19,315	71.01	60.25
双方向言語モデル	200	4,979	71.21	60.44
双方向言語モデル	250	1,488	71.44	60.90
双方向言語モデル	300	482	71.42	60.73

られたが，ベースラインの SARI を上回ることはなかった．

4.3.3 流暢性に関する手法の閾値

流暢性に関する手法について，閾値を変更しながらパラレルコーパスフィルタリングを実施し，検証データにおけるテキスト平易化の性能を評価した結果を表 5 に示す．

単方向言語モデルに基づく手法では，パープレキシティが 100 を超える文を含む文対を除去することで検証データにおける最高の SARI が得られたものの，ベースラインの性能を上回らなかった．また，双方向言語モデルに基づく手法では，パープレキシティが 250 を超える文を含む文対を除去することで検証データにおける最高の SARI が得られたものの，ベースラインの性能を上回らなかった．

4.4 実験結果

検証データにおいて最高の SARI を達成した閾値を用いて，評価データにおける各手法の性能を評価した．表 6 に実験結果を示す．なお，表 6 において太字はベースラインを上回る性能を表し，下線は最高性能を表している．

実験の結果，文長差および流暢性に関するパラレルコーパスフィルタリングによって，BLEU と SARI の両方が改善できた．一方で，同義性に関するパラレルコーパスフィルタリングでは，BLEU と SARI の両方を悪化させる結果となった．特に，トークン数の差に基づく手法が有効であり，単語数の差に基づく手法がベースラインを 2.04 ポイント上回る最高の BLEU を達成し，文字数の差に基づく手法がベースラインを 2.21 ポイント上回る最高の SARI を達成した．

4.5 考察

本実験の結果から，やさしい日本語コーパスにおいては，難解文と平易文の間の文長差および平易文の流暢性に基づくパラレルコーパスフィルタリングが有効であることがわかった．これは，やさしい日本語コーパスでは「平易な 2,000 単語を用いて記述する」という制約のために，表 2 に示したような無理な平易化が行われているからだと考えられる．難解な単語を避けた結果，その単語を説明するような表現へ置換している例が頻繁に見られた．これによって，難解文と平易文の間で文長に顕著な差が見られたり，平易文の流暢性が損なわれたりしている．提案手法では，これらのノイズとなる文対を訓練データから適切に除去し，テキスト平易化モデルの品質を改善できた．

同義性に関するパラレルコーパスフィルタリングが有効に機能しなかった要因は，2 つ挙げられる．1 点目は，単語や文の分散表現が細かな意味の違いを正確に捉えられないことである．例えば，単語分散表現や文分散表現に基づく手法において，類似度が 0.5 より小さい，つまり同義性が低いと判断された文対に，「本当に瓜二つだわ」と「本当にとても似ている」があった．これは実際には妥当な訓練事例の文対であるが，本研究では訓練データから除去され

表 6：評価データにおけるパラレルコーパスフィルタリングの性能評価

手法	BLEU	SARI
ベースライン	81.56	62.88
文長差：トークン数の差（文字）	**82.31**	**65.09**
文長差：トークン数の差（単語）	**83.60**	**63.69**
文長差：トークン数の差（サブワード）	80.76	**63.65**
文長差：編集距離（文字）	**81.98**	**63.69**
文長差：編集距離（単語）	**83.38**	**63.13**
文長差：編集距離（サブワード）	**83.20**	**62.93**
同義性：平均単語ベクトル	80.68	59.80
同義性：単語アライメント	80.93	62.31
同義性：文ベクトル	81.50	61.32
流暢性：単方向言語モデル	**82.34**	**63.00**
流暢性：双方向言語モデル	**82.15**	**63.05**

てしまった．2 点目は，そもそも大きく意味の異なる文対が少なかったということである．やさしい日本語コーパスは人手で平易文を記述したコーパスであるため，意味が大きく離れてしまうような文対は生まれにくかったと考えられる．以上の分析から，対象コーパスの性質を考慮したうえでパラレルコーパスフィルタリングの手法を設計することが重要であると言える．

5. おわりに

本研究では，日本語におけるテキスト平易化モデルの性能改善のために，やさしい日本語コーパスの訓練データから，文長差・同義性・流暢性のそれぞれの観点からノイズとなる文対を除去するパラレルコーパスフィルタリングの手法を提案した．BART に基づくテキスト平易化モデルを対象とする評価実験の結果，文長差に関するパラレルコーパスフィルタリングによって，訓練データの全体を用いて訓練するベースラインよりも BLEU および SARI の評価を改善できた．提案手法では，無理な平易化によって冗長な表現となった平易文を適切に検出し，パラレルコーパスフィルタリングに活用できたと考えられる．

今後の課題としては，効果があった複数のパラレルコーパスフィルタリング手法を組み合わせてテキスト平易化の品質を更に改善すること，英語などの他言語におけるテキスト平易化タスクやスタイル変換などの類似タスクへ適用すること，日本語における文間意味的類似度推定の性能を改善することなどに取り組んでいきたい．

謝辞

本研究はJSPS科研費（基盤研究B，課題番号：JP22H03651）および国立研究開発法人情報通信研究機構の委託研究（課題番号：225）による助成を受けたものです．

参考文献

[1] 岩田一成. 言語サービスにおける英語志向：「生活のための日本語: 全国調査」結果と広島の事例から. 社会言語科学, Vol. 13, No. 1, pp. 81-94, 2010.
[2] 梶原智之, 小町守. 平易なコーパスを用いないテキスト平易化. 自然言語処理,Vol. 25, No. 2, pp. 223-249, 2018.
[3] 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.
[4] 中町礼文, 梶原智之. 事前訓練済み系列変換モデルに基づくやさしい日本語への平易化. 情報処理学会第 83 回全国大会, pp. 607-608, 2021.
[5] 山本和英, 丸山拓海, 角張竜晴, 稲岡夢人, 小川耀一朗, 勝田哲弘, 髙橋寛治.やさしい日本語対訳コーパスの構築. 言語処理学会第 23 回年次大会, pp.763-766, 2017.
[6] Takumi Maruyama and Kazuhide Yamamoto. Simplified Corpus with Core Vocabulary. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, 2018.
[7] Akihiro Katsuta and Kazuhide Yamamoto. Crowdsourced Corpus of Sentence Simplification with Core Vocabulary. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, 2018.
[8] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871-7880, 2020.
[9] Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. Data-Driven Sentence Simplification: Survey and Benchmark. Computational Linguistics, pp. 135–187, 2020.
[10] William Coster and David Kauchak. Simple English Wikipedia: A New Text Simplification Task. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 665–669, 2011.
[11] Lucia Specia. Translating from Complex to Simplified Sentences. In Proceedings of the 9th International Conference on Computational Processing of the Portuguese Language, pp. 30-39, 2010.
[12] Sander Wubben, Antal van den Bosch, and Emiel Krahmer. Sentence Simplification by Monolingual Machine Translation. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pp. 1015–1024, 2012.
[13] Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. Exploring Neural Text Simplification Models. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 85–91, 2017.
[14] Xingxing Zhang and Mirella Lapata. Sentence Simplification with Deep Reinforcement Learning. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 584–594, 2017.
[15] Tomoyuki Kajiwara. Negative Lexically Constrained Decoding for Paraphrase Generation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6047–6052, 2019.
[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In Advances in Neural Information Processing Systems 30, pp. 5998–6008, 2017.

[17] Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. Integrating Transformer and Paraphrase Rules for Sentence Simplification. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3164–3173, 2018.
[18] Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. Controllable Sentence Simplification. In Proceedings of the 12th Language Resources and Evaluation Conference, pp. 4689–4698, 2020.
[19] Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes and Benoît Sagot. MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases. Proceedings of the 13th Language Resources and Evaluation Conference, pp. 1651-1664, 2022.
[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186, 2019.
[21] William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. Aligning Sentences from Standard Wikipedia to Simple Wikipedia. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 211–217, 2015.
[22] Tomoyuki Kajiwara and Mamoru Komachi. Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment between Word Embeddings. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1147–1158, 2016.
[23] Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. Neural CRF Model for Sentence Alignment in Text Simplification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7943–7960, 2020.
[24] Tomoyuki Kajiwara, Biwa Miura, and Yuki Arase. Monolingual Transfer Learning via Bilingual Translators for Style-Sensitive Paraphrase Generation. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, pp. 8042–8049, 2020.
[25] Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. Findings of the WMT 2020 Shared Task on Parallel Corpus Filtering and Alignment. In Proceedings of the Fifth Conference on Machine Translation, pp. 726–742, 2020.
[26] Reno Kriz, João Sedoc, Marianna Apidianaki, Carolina Zheng, Gaurav Kumar, Eleni Miltsakaki, and Chris Callison-Burch. Complexity-Weighted Loss and Diverse Reranking for Sentence Simplification. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 3137–3147, 2019.
[27] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, Vol. 5, pp. 135–146, 2017.
[28] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. Universal Sentence Encoder for English. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demon strations, pp. 169–174, 2018.
[29] Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 440–450, 2018.
[30] Yangqiu Song and Dan Roth. Unsupervised Sparse Vector Densification for Short Text Similarity. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.1275–1280, 2015.
[31] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.
[32] Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. Masked Language Model Scoring. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 2699–2712, 2020.
[33] Muthu Chidambaram, Yinfei Yang, Daniel Cer, Steve Yuan, Yunhsuan Sung, Brian Strope, and Ray Kurzweil. Learning Cross-Lingual Sentence Representations via a Multi-task Dual-Encoder Model. In Proceedings of the 4th Workshop on Representation Learning for NLP, pp. 250–259, 2019.
[34] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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.
[35] Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing Statistical Machine Translation for Text Simplification. Transactions of the Association for Computational Linguistics, Vol. 4, pp. 401–415, 2016.
[36] Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. EASSE: Easier Automatic Sentence Simplification Evaluation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th Inter-national Joint Conference on Natural Language Processing: System Demonstrations, pp. 49–54, 2019.
[37] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying Conditional Random Fields to Japanese Morphological Analysis. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 230–237, 2004.
[38] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 48–53, 2019.
[39] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, pp. 1-15, 2015.
[40] Hajime Morita, Daisuke Kawahara, and Sadao Kurohashi. Morphological Analysis for Unsegmented Languages using Recurrent Neural Network Language Model. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2292–2297, 2015.
[41] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 66–71, 2018.
[42] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45, 2020.