Text Simplification without Simplified Corpora

TOKYO METROPOLITAN UNIVERSITY

Tomoyuki Kajiwara

Department of Information and Communication Systems
Graduate School of System Design
Tokyo Metropolitan University

March 22, 2018

A Doctoral Dissertation
submitted to Graduate School of System Design,
Tokyo Metropolitan University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Tomoyuki Kajiwara

Thesis Committee:

Mamoru Komachi	Associate Professor
Toru Yamaguchi	Professor
Yasufumi Takama	Professor
Naoaki Okazaki	Professor

Text Simplification without Simplified Corpora*

Tomoyuki Kajiwara

Abstract

Text simplification is the task of rewriting complex text into a simpler form while preserving its meaning. Systems that automatically pursue this task can potentially be used for assisting reading comprehension of less language-competent people, such as learners and children. Such systems would also improve the performance of other Natural Language Processing applications. As with machine translation and abstractive summarization, this task is positioned as a Text-to-Text Generation task in natural language processing.

Current work has two approaches: lexical substitution and monolingual translation. In the former, a simpler synonymous sentence is generated by the pipeline of complex word identification, substitution generation, and substitution ranking. In the latter, a simpler synonymous sentence is generated using machine translation tools. In both approaches, mainstream methods acquire simplification rules from a large-scale parallel corpus. Therefore, text simplification was studied mainly in English for where rich resources are available. However, a large-scale simplified corpus for text simplification cannot be used in many language other than English.

In this research, we propose text simplification methods by lexical substitution approach and monolingual translation approach for languages that cannot use large-scale simplified corpora, especially Japanese. As a lexical substitution approach without simplified corpora, we propose novel paraphrase acquisition, meaning preservation filtering, simplicity filtering, and grammaticality ranking methods for Japanese. In addition, as a monolingual translation approach without simplified corpora, we construct a pseudo-parallel corpus for text simplification from a raw corpus using readability assessment and sentence alignment, and enable text simplification using machine translation tools in any language.

*Doctoral Dissertation, Department of Information and Communication Systems, Graduate School of System Design, Tokyo Metropolitan University, March 22, 2018.

Experimental results show that our lexical substitution approach outperforms the previous language-independent unsupervised method. Moreover, in the monolingual translation approach, the experimental results show that our pseudo-parallel corpus succeeds in training machine translation tools as well as existing parallel corpora for text simplification.

Keywords:

Natural Language Processing, Text Simplification, Paraphrase Acquisition, Semantic Textual Similarity, Quality Estimation

平易なコーパスを用いないテキスト平易化*

梶原 智之

内容梗概

テキスト平易化は、難解なテキストの意味を保持したまま平易に書き換えるタスクである。システムは、言語学習者や子どもをはじめとする人々の文章読解を支援し、他の自然言語処理応用タスクの性能改善にも寄与する。このタスクは、機械翻訳や文書要約などと同じく、自然言語処理におけるテキストからのテキスト生成タスクとして位置づけられる。

先行研究には語彙的換言と単言語翻訳の２つのアプローチがある。語彙的換言アプローチでは「難解語検出・換言生成・ランキング」のパイプラインで平易な同義文を生成する。単言語翻訳アプローチでは機械翻訳器を用いて平易な同義文を生成する。どちらのアプローチでも大規模なパラレルコーパスから平易化規則を獲得する手法が主流であるため、これまでは言語資源の豊富な英語を中心に研究されてきた。しかし、英語以外の多くの言語では平易に書かれた大規模コーパスを利用できない問題がある。

本研究では、平易な大規模コーパスを利用できない言語、特に日本語を対象として、語彙的換言アプローチと単言語翻訳アプローチによるテキスト平易化を実現する。まず、平易なコーパスを用いない語彙的換言アプローチとして、本研究では日本語のための新しい言い換え知識獲得、意味的等価性フィルタリング、平易性フィルタリング、文法性ランキングの各手法を提案する。また、平易なコーパスを用いない単言語翻訳アプローチとして、本研究では文の難易度推定と文間類似度推定を組合せて生コーパスからテキスト平易化のための疑似パラレルコーパスを構築し、任意の言語での機械翻訳器を用いたテキスト平易化を可能にする。

実験結果によって、我々の語彙的換言アプローチが既存の言語非依存な教師なし手法を上回ることを示す。また、単言語翻訳アプローチにおいては、我々の疑似パラレルコーパスが既存のテキスト平易化のためのパラレルコーパスと同等に機械翻訳器の訓練を成功させることを示す。

*首都大学東京 システムデザイン研究科 情報通信システム学域 博士論文, 2018 年 3 月 22 日.

キーワード

自然言語処理, テキスト平易化, 言い換え知識獲得, 意味的文間類似度, 品質推定

Acknowledgements

博士課程の 3 年間ご指導いただきました小町守先生に深く感謝いたします。直接の研究指導はもちろんのこと、インターンシップや共同研究として多くの研究者に学ぶ機会を与えていただき、研究者としての幅を広げることができました。

博士論文の審査を引き受けてくださいました山口亨先生、高間康史先生、岡崎直観先生に感謝いたします。副査の先生方には、お忙しい中、丁寧に博士論文を見ていただき、様々な観点からご指導をいただきました。

長岡技術科学大学の山本和英先生には、研究室配属から修士課程までの 3 年間ご指導いただきました。研究の基本とプレゼンテーションについて丁寧にご指導いただき、初めての学会発表であった NLP 若手の会で奨励賞を受賞することができました。6 年間の研究生活は順調なことばかりではありませんでしたが、この最初の成功体験のおかげでここまで頑張って来られたと思います。

富士通研究所の潮田明さん、大倉清司さんには、学士課程 4 年次に初めてのインターンシップを経験させていただきました。また、富士秀さん、岩倉友哉さんにはインターンシップ後も様々な機会にアドバイスをいただきました。

ブレインパッドの太田満久さんには、修士課程 1 年次にインターンシップでお世話になりました。就職活動をするか博士課程に進学するか迷っていた時期でしたが、博士号取得後にエンジニアとして活躍されている太田さんの姿を見て博士課程への進学を決心しました。

博士課程 1 年次には、リバプール大学の客員研究員として Danushka Bollegala 先生にご指導いただきました。一緒にご指導いただいた国立情報学研究所の河原林健一先生、吉田悠一先生にも感謝いたします。博士課程での以降の研究も、多くはこのとき学んだことからヒントを得て着想しました。論文の書き方についても、自分が書いた初稿と先生に直してもらった原稿とを何度も見比べ、多くのことを学びました。博士課程の早い時期に、研究の進め方や論文の書き方をトップレベルの研究者からご指導いただくことができ、本当にありがたい機会でした。

博士課程 2 年次からは、統計数理研究所の持橋大地先生にご指導いただきました。単にタスクの性能を高めるだけではなく、言語現象を数学で説明するという研究の姿勢について学びました。

博士課程 3 年次には、情報通信研究機構の協力研究員として藤田篤さんにご指導いただきました。藤田さんには毎日ミーティングの時間を取っていただき、多くのことを学ぶことができました。研究以外の話題にも親身になって相談に乗っていただきました。心より感謝いたします。「足場を固めながら進む」と繰り返しご指導いただき、研究が進むということとタスクの性能が上がるということが必ずしも同義ではないということを理解しました。何を明らかにしたいのか、何が明らかになったのか、ということを意識し、藤田さんのように自分に厳しく真摯に研究に取り組んでいきたいと思います。温かく受け入れてくださり、多くのご助言をいただきました先進的翻訳技術研究室の皆様にも感謝いたします。

指導教員や共著者の皆様以外にも、多くの研究者の方々にお世話になりました。トークに招待いただきご助言をいただきました東北大学の乾健太郎先生、愛媛大学の二宮崇先生、LINE の佐藤敏紀さん、明石高専の奥村紀之先生、大阪大学の荒瀬由紀先生、奈良先端科学技術大学院大学の能地宏先生に感謝いたします。また、Aim4ACL にて国際会議に投稿予定の研究や原稿に対してご助言をいただきました先生方に感謝いたします。そして、NLP 東京 D の会を一緒に立ち上げ、議論してくださった同年代の博士課程の皆様に感謝いたします。NLP 東京 D の会の皆様のおかげで、博士課程の 3 年間を楽しく過ごすことができました。

首都大学東京の小町研究室および長岡技術科学大学の山本研究室で一緒に過ごした学生の皆様にも感謝いたします。皆様との日々の議論のおかげで、研究を進めてくることができました。特に、自然言語処理の研究を始めるきっかけを与えてくださった真嘉比愛さん、一緒に研究を進めてくださった鈴木由衣さん、Aizhan Imankulova さん、金子正弘さん、小平知範さん、関沢祐樹さん、塩田健人さん、野口真人さん、大森光さん、嶋中宏希さん、どうもありがとうございました。

また、宇摩剣道連盟の先生方および四国中央剣道会の後輩たちにも学生生活を支えていただきました。研究が上手くいかないときもありましたが、後輩剣士の皆さんが成長している姿を見せてくれたり、試合での活躍を聞かせてくれたおかげで、私はいつも幸せに過ごすことができました。

最後に、長い学生生活を応援し続けてくれた両親、妹、祖父母に深く感謝し、謝辞といたします。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Text simplification is the task of rewriting complex text into a simpler form while preserving its meaning. Systems that automatically pursue this task can potentially be used for assisting reading comprehension of less language-competent people, such as learners [88] and children [11]. Such systems would also improve the performance of other natural language processing tasks, such as information extraction [30] and machine translation [115].

Text simplification is one on the Text-to-Text Generation tasks with machine translation, paraphrase generation, abstractive summarization, and error correction. Machine translation transforms input sentence into different language sentence, while text simplification transforms it into same language sentence. The degree of meaning preservation differs between paraphrase generation and this task. Unlike paraphrase generation, this task often deletes unnecessary expressions. Although it is common with abstractive summarization in terms of deleting unnecessary expressions, text simplification often adds detailed explanation to assist the reader's reading comprehension. Moreover, while error correction improves grammaticality of input sentence, text simplification improves its simplicity.

Text simplification includes two subtasks [94]: lexical simplification and syntactic simplification. Lexical simplification substitutes complex words/phrases in input sentence into a simpler words/phrases. Syntactic simplification transforms complex structures of input sentence into a simpler structures, such as split into shorter sentences or reordering. We work on lexical simplification subtask which deals only with translation from one sentence to one sentence. Lexical simplification has two major approaches: lexical simplification approaches [28, 11, 124, 16, 44, 34, 83, 80] and monolingual translation approaches [98, 128, 27, 26, 120, 112, 35, 127]. Previously,

text simplification was studied mainly in English for where rich resources are available such as a manually constructed text simplification corpus [121], a large-scale simplified corpus (Simple English Wikipedia[1]), and a paraphrase database [85]. However, improving the English model with abundant resources cannot benefit from text simplification in other languages with poor resources. In this thesis, we simplify sentences without simplified corpora for Japanese.

1.1. Main Contributions

1. We propose novel, state-of-the-art strategies for Japanese lexical simplification. For three types of candidate acquisition, four types of meaning preservation filtering, four types of simplicity filtering, and three types of grammaticality ranking, we comprehensively experiment and build a state-of-the-art Japanese lexical simplification system.

2. We build a first evaluation dataset for Japanese lexical simplification. This dataset enables a step-by-step automatic evaluation and an overall automatic evaluation of the simplification pipeline.

3. We propose to use sentence similarity based on alignment between word embeddings for text simplification. In both lexical substitution approach and monolingual translation approach, a monolingual parallel corpus is indispensable for simplification rule acquisition. Our sentence similarity measure outperforms previous works in alignment task of complex and simple sentences.

4. By improving sentence alignment, we achieve the best performance of English text simplification model using PBSMT. This experimental result makes us reconfirm the fact that better data help to develop a better model.

5. For text simplification in languages that cannot use large-scale simplified corpora, we build a pseudo-parallel corpus from a raw corpus using readability assessment and sentence alignment. Experimental results show that our pseudo-parallel corpus can simplify as good as using large-scale simplified corpora.

6. We combine the paraphrasability score from monolingual corpora and from bilingual corpora to propose a novel paraphrasability score. Levy and Goldberg [65]

[1] http://simple.wikipedia.org/

explained a well-known representation learning method for word embeddings, the skip-gram with negative sampling [73], as a matrix factorization of a word-context co-occurrence matrix with shifted positive PMI. In this work, we explained a well-known method for paraphrase acquisition, bilingual pivoting [10], as an unsmoothed version of PMI.

7. In order to further improve sentence alignment, we propose a domain adaptation method for sentence similarity. Experimental results show that updating the general word similarity with the word similarity specialised for a given corpus improves the sentence similarity based on word alignment.

8. We propose a novel quality estimation method for text simplification using our proposed sentence similarity measures based on word alignment. In text simplification, since automatic evaluation metrics using single reference have low correlation with manual evaluation [122], quality estimation, i.e., automatic evaluation without reference, has been drawing much attention [117]. As a result of experiments, we confirm that our alignment-based features computed on the basis of word embeddings and paraphrase lexicons can achieve the state-of-the-art performance.

1.2. Structure of the Thesis

Chapter 2 presents our approach to Japanese lexical simplification. We build Japanese lexical simplification system (Contribution 1) and evaluation dataset (Contribution 2).

Chapter 3 presents our approach to English and Japanese sentence simplification. We investigate the best sentence alignment method for text simplification (Contribution 3) and build the state-of-the-art simplification model based on PBSMT (Contribution 4). In addition, it shows that pseudo-parallel corpus obtained from a raw corpus by readability assessment and the sentence alignment is as effective as parallel corpus, and it opens the door to multilingualization of text simplification (Contribution 5).

Chapter 4 presents experimental results in English for our three works to further improve text simplification. Section 4.1 considers both monolingual corpus and bilingual corpus to acquire paraphrase lexicon more accurately (Contribution 6). Section 4.2 proposes a domain adaptation method to calculate sentence similarity more accurately (Contribution 7). Section 4.3 describes a quality estimation method using sentences

similarity based on word alignment for more accurate automatic evaluation (Contribution 8).

Finally, in Chapter 5 we provide our final remarks and directions for future work.

Chapter 2

Lexical Simplification without Simplified Corpora

In this section, we perform text simplification using lexical substitution approach in Japanese. Similar to previous works [94, 80], we paraphrase complex word in context into simpler version according to the following procedure.

1. Candidate Acquisition
2. Meaning Preservation Filtering
3. Simplicity Filtering
4. Grammaticality Ranking

In Japanese, parallel corpora to acquire simplification rules cannot be used. Moreover, automatic evaluation is difficult becaus there is no evaluation dataset for Japanese lexical simplification.

First, in Section 2.1, we acquire paraphrases as simplification candidates. Next, in Section 2.2, we remove pairs with low likelihood among the paraphrase pairs. Moreover, in Section 2.3, we extract only paraphrase pairs from complex to simple words. Finally, in Section 2.4, we select the paraphrase suitable for the context of the input sentence.

In addition, Section 2.6 evaluates each lexical simplification method on our evaluation dataset constructed in Section 2.5.

2.1. Candidate Acquisition

In Japanese that cannot use large-scale simplified corpora, simplification rule acquisition from parallel corpus [44] cannot be used. Moreover, paraphrase acquisition from monolingual corpora [13, 69] using distributional similarity [38] is difficult to discriminate between synonym and antonym [77].

In Section 2.1.1, we forcus on the definition statements as a paraphrase acquisition source in place of a monolingual corpus and a parallel corpus. In Section 2.1.2, we integrate multiple synonym dictionaries constructed manually for high-quality paraphrasing. In Section 2.1.3, we acquire paraphrases using word alignment on bilingual corpus for large-scale paraphrasing.

2.1.1 Simplification Rules from Definition Statements

The Japanese dictionary is a resource that explains headwords by definition statements. Therefore, the following two characteristics can be assumed.

1. Corresponding headword and definition statement are semantically equivalent.

2. Definition statements are written in easier words than headwords for users to read easily.

We use these characteristics to acquire simplification rules through extracting synonymous expressions of headwords from definition statements.

Since Japanese is a head-final language, Kaji et al. [51] proposed a method to acquire paraphrase of headwords from the end of definition statements. However, paraphrase of the headwords does not appear only at the end of definition statements.

Therefore, we widely collect paraphrase candidates of headwords from the whole definition statements. In order to reduce noise, we use constraints of part of speech [83] and target only words having the same part of speech as the headword. We acquire paraphrase candidates of headwords from definition statements in the following procedure.

1. Morphologically analyze definition statements.

2. Acquire all content words with the same part of speech as the headword.

2.1.2 Manually Acquired Paraphrase Lexicon

In the previous section, we proposed a method to acquire simplification rules by considering pairs of headwords and definition statements as pairs of complex and simple texts. As with simplification rule acquisition method from parallel corpus for text simplification [44], this method has the advantage that difficulty estimation of paraphrase pair is not required but its performance is affected by alignment accuracy.

In this section, we integrate five types of Japanese synonym dictionaries constructed manually for high-quality paraphrasing.

Lexical Paraphrase Dictionary of Japanese Content Words[1] **[123]**

In this dictionary, paraphrases are given manually by nouns, sahen-nouns, verbs, adjectives, adverbs among headwords of morpheme dictionaries in morphological analyzeer JUMAN (Ver.7.0) [59]. This dictionary allows for missing information and include irreversible transformations such as "canary" $\rightarrow$ "bird". There are 25,503 paraphrases based on phrases of up to three words.

Japanese WordNet Synonyms Database[2] **(Ver.1.0)**

This is a collection of 11,753 synonym pairs, which were collected using synsets in Japanese WordNet [49]. Word pairs were created using words in a synset, which is a cluster of words that share the same sense, and were manually annotated. The word pairs that were manually annotated as synonym pairs were included in the database.

Verb Entailment Database[3] **(Ver.1.3.1)**

This is a collection of automatically acquired [66, 119, 108, 39, 40] verb pairs. Eight types of labels such as entailment, presupposition, and action-reaction are manually given to these verb pairs. We use 94,025 verb pairs classified as entailment.

Database of Japanese Orthographic Variant Pairs[4] **(Ver.1.1)**

This is a collection of noun phrase pairs with edit distance of 1. Ten types of labels such as synonym, allograph, and erratum are manually given to these noun phrase pairs. We use 50,825 noun phrase pairs classified as synonym or erratum.

[1] http://www.jnlp.org/SNOW/D2

[2] http://compling.hss.ntu.edu.sg/wnja/jpn/downloads.html

[3] https://alaginrc.nict.go.jp/resources/nict-resource/li-info/li-outline.html#A-2

[4] https://alaginrc.nict.go.jp/resources/nict-resource/li-info/li-outline.html#A-7

Case Base for Basic Semantic Relations[5] **(Ver.1.4)**

This is a collection of word pairs with high contextual similarity included in the database of similar context terms[6]. Eight types of labels such as variant, abbreviation, and synonym are manually given to these word pairs. We use 78,260 word pairs classified as variant, abbreviation, or synonym.

2.1.3 Automatically Acquired Paraphrase Lexicon

In the previous section, we used synonym dictionaries constructed manually to acquire simplification rules. However, building a large-scale and high-quality synonym dictionary requires a large cost. Therefore, it is difficult to keep up with new words or new meaings. Moreover, it is also difficult to expand from word to phrase.

In this section, we acquire paraphrases using word alignment on bilingual corpus for large-scale paraphrasing. By applying phrase tabel acquire [79] in phrase-based statistical machine translation, Bannard and Callison-Burch [10] proposed a method (Bilingual Pivoting) to acquire large-scale paraphrases from a bilingual corpus using foreign language phrase as a pivot. In other words, using two phrase tables of $j_1 \rightarrow e$ and $e \rightarrow j_2$, acquire two Japanese phrases $< j_1, j_2 >$ as a paraphrase pair via an English phrase e.

2.2. Meaning Preservation Filtering

In this section, we remove pairs with low likelihood among the paraphrase pairs acquired in Section 2.1.

To estimate semantic equivalence between words, there are methods based on semantic similarity and paraphrase probability. In Section 2.2.1, we use the distance of the path on WordNet [14, 37] which is a classic method for estimating semantic similarity between words. In Section 2.2.2, we use the cosine similarity between word embeddings which is the de facto standard method for estimating semantic similarity between words. In Section 2.2.3, we estimate the paraphrase probability between words using word alignment probability on bilingual corpus which is the de facto standard method for paraphrase acquisition. In Section 2.2.4, we estimate the equivalence

[5] https://alaginrc.nict.go.jp/resources/nict-resource/li-info/li-outline.html#A-9
[6] https://alaginrc.nict.go.jp/resources/nict-resource/li-info/li-outline.html#A-1

of meaning between words using pointwise mutual information which smoothes the paraphrase probability of Section 2.2.3 with word probability.

2.2.1 Path Distance Similarity

Thesauri such as WordNet [75, 49] are resources classifying words from the viewpoint of semantic hypernym-hyponym relations. The closer the distance between words on the thesaurus, the higher the semantic similarity between words. Path distance similarity has been used in many previous works [14, 37] as a semantic similarity estimation method between words based on knowledge base. The semantic similarity $\mathrm{PDS}(j_1, j_2)$ between words j_1 and j_2 is calculated as follows using the depth of each word on the thesaurus d_w and the depth of the hypernym common to both words d_c.

$$\mathrm{PDS}(j_1, j_2) = \frac{2d_c(j_1, j_2)}{d_w(j_1) + d_w(j_2)} \tag{2.1}$$

2.2.2 Context Similarity

Based on the distributional hypothesis [38], semantic similarity between words can be estimated using the similarity of the distribution of words co-occurring as context. Context similarity has been used in many previous works [25, 34] as a semantic similarity estimation method between words based on corpus. Especially, methods using word embeddings [73, 87, 64] which can be constructed from monolingual corpora by unsupervised learning are the de facto standard method for estimating semantic similarity between words. The semantic similarity $\mathrm{WES}(j_1, j_2)$ between words j_1 and j_2 is calculated as follows using the cosine similarity between word embeddings $\vec{j_1}$ and $\vec{j_2}$.

$$\mathrm{WES}(j_1, j_2) = \cos(\vec{j_1}, \vec{j_2}) \tag{2.2}$$

2.2.3 Alignment Probability

Bilingual pivoting [10], described in Section 2.1.3, employs a conditional paraphrase probability $p(j_2 \mid j_1)$ as a paraphrasability measure, when there are word alignments between a Japanese phrase j_1 and an English phrase e, and between the English phrase

e and another Japanese phrase j_2 on a bilingual corpus. It calculates the probability from an Japanese phrase j_1 to another Japanese phrase j_2 using word alignment probabilities $p(e \mid j_1)$ and $p(j_2 \mid e)$; here, the English phrase e is used as the pivot.

$$\begin{aligned} p(j_2 \mid j_1) &= \sum_e p(j_2 \mid e, j_1)\, p(e \mid j_1) \\ &\approx \sum_e p(j_2 \mid e)\, p(e \mid j_1) \end{aligned} \tag{2.3}$$

It assumes conditional independence of j_1 and j_2 given e so that the last equation can be estimated easily using phrase-based statistical machine translation models. One of the advantages is that it requires only two translation models to estimate paraphrasability. However, since the conditional probability is asymmetric, it may introduce irrelevant paraphrases that do not hold the same meaning as the original one.

To mitigate this, PPDB[7] [33] defined the symmetric paraphrase score $\mathrm{BP}(j_1, j_2)$ using bi-directional bilingual pivoting.

$$\mathrm{BP}(j_1, j_2) = -\lambda_1 \log p(j_2 \mid j_1) - \lambda_2 \log p(j_1 \mid j_2) \tag{2.4}$$

In this study, without loss of generality, we set[8] $\lambda_1 = \lambda_2 = -1$.

$$\mathrm{BP}(j_1, j_2) = \log p(j_2 \mid j_1) + \log p(j_1 \mid j_2) \tag{2.5}$$

2.2.4 MIPA Score

The bi-directional bilingual pivoting of PPDB [33] constrains paraphrasability to be strictly symmetric. However, though it is extremely good at extracting synonymous expressions, it tends to give high scores to frequent but irrelevant phrases since bilingual pivoting itself contains noisy phrase pairs due to word alignment errors.

To address the problem of frequent phrases, we smooth paraphrasability by bilingual pivoting in Equation (2.5) using word probabilities $p(j_1)$ and $p(j_2)$ from a monolingual corpus that is sufficiently larger than the bilingual corpus.

$$\mathrm{BPMI}(j_1, j_2) = \log p(j_2 \mid j_1) + \log p(j_1 \mid j_2) - \log p(j_1) - \log p(j_2) \tag{2.6}$$

By doing so, we can interpret the bi-directional bilingual pivoting as an unsmoothed version of PMI. Since the difference of the logarithms of the numerator and denominator is equal to the logarithm of the quotient, we can transform Equation (2.6) as

[7] http://www.cis.upenn.edu/~ccb/ppdb/

[8] Similar to PPDB ($\lambda_1 = \lambda_2 = 1$), we have two equally weighted components.

follows.

$$\begin{aligned} \mathrm{BPMI}(j_1, j_2) &= \log \frac{p(j_2 \mid j_1)}{p(j_2)} + \log \frac{p(j_1 \mid j_2)}{p(j_1)} \\ &= 2\mathrm{PMI}(j_1, j_2) \end{aligned} \tag{2.7}$$

since we can transform PMI into the following forms using Bayes' theorem.

$$\begin{aligned} \mathrm{PMI}(x,y) &= \log \frac{p(x,y)}{p(x)p(y)} \\ &= \log \frac{p(y \mid x)p(x)}{p(x)p(y)} = \log \frac{p(y \mid x)}{p(y)} \\ &= \log \frac{p(x \mid y)p(y)}{p(x)p(y)} = \log \frac{p(x \mid y)}{p(x)} \end{aligned} \tag{2.8}$$

In low-frequency word pairs, it is well-known that PMI becomes unreasonably large because of coincidental co-occurrence. In order to avoid this problem, Evert [31] proposed Local PMI that assigns weights to PMI depending on the co-occurrence frequency of word pairs.

$$\mathrm{LPMI}(x,y) = n(x,y) \cdot \mathrm{PMI}(x,y) \tag{2.9}$$

In this study, however, it is difficult to directly calculate the weight corresponding to $n(x,y)$ in Equation (2.9) on the bilingual corpus. Furthermore, what we want to calculate is not the strength of co-occurrence (relation) between words, but paraphrasability between words. Therefore, it is not appropriate to count the co-occurrence frequency on a monolingual corpus like Local PMI.

Alternatively, we use as a weight the distributional similarity, which is often used as a paraphrase acquisition from a monolingual corpus [25, 34].

$$\mathrm{MIPA}(j_1, j_2) = \cos(\vec{j_1}, \vec{j_2}) \cdot \mathrm{BPMI}(j_1, j_2) \tag{2.10}$$

Equation (2.10) simultaneously considers paraphrasability based on the monolingual corpus (distributional similarity) and paraphrasability based on the bilingual corpus (bilingual pivoting). Distributional similarity is robust against noise associated with unrelated word pairs as opposed to bilingual pivoting. Bilingual pivoting is robust to noise arising from antonym pairs unlike distributional similarity. Therefore, $\mathrm{MIPA}(j_1, j_2)$ can estimate paraphrasability robustly by complementing the disadvantages.

2.3. Simplicity Filtering

In this section, we assign difficulty to each word in paraphrase pairs with high likelihood acquired in Section 2.2, and extract paraphrase pairs from complex to simple words.

2.3.1 Word Frequency

In SemEval-2012 English Lexical Simplification Task [99] of reordering word lists form the viewpoint of simplicity, word frequency as baseline achieved the second place on the 12 system and showed the effectiveness of paraphrasing to the high frequency word in the lexical simplification task. We also define more frequent words as simpler words.

2.3.2 Word Familiarity

Word Familiarity [7] is a score that expresses how well a word is known as a real number from 1 (unknown) to 7 (well known). According to previous work [48], we define words with higher familiarity score as simpler words.

2.3.3 JLPT Simplicity

JLPT is a Japanese language proficiency test for non-native Japanese speakers. The criterion for that question is to classify each word in Japanese into four levels from 1st grade (complex) to 4th grade (simple). We define words with higher JLPT grade as simpler words.

2.3.4 JEV Difficulty

Japanese educational vocabulary[9] (Ver.1.0) [105] is a word list based on vocabulary analysis of balanced corpus of contemporary written Japanese [67] and Japanese textbook corpus (100 Japanese textbooks on the market). Japanese teachers gave each Japanese word six levels of difficulty. We define words with lower JEV level as simpler words.

[9] http://jhlee.sakura.ne.jp/JEV.html

2.4. Grammaticality Ranking

In this section, we select the paraphrase suitable for the context of the input sentence using simplification rules acquired in Section 2.3.

2.4.1 Language Model Probability

We calculate the N-gram language model probability LM_N of the input sentence and the paraphrased sentence, and select the sentence with the highest likelihood. In this way, the most fluent simplification rule can be applied considering the context of N words before and after the target word.

$$\begin{aligned}\mathrm{LM}_N(w_1, w_2, ..., w_n) &= \prod_{k=1}^{n} p(w_k \mid w_{k-N+1}^{k-1}) \\ &= p(w_1)\, p(w_2 \mid w_1)\, p(w_3 \mid w_1, w_2) ... p(w_n \mid w_{n-N+1}, ..., w_{n-1})\end{aligned} \tag{2.11}$$

2.4.2 Context Embedding Similarity

In lexical substitution task, candidate ranking methods [72, 8] based on cosine similarity between context words and paraphrase candidate are proposed. We select a paraphrase that fits the context as follows using the cosine similarity cos between context word $c \in C$ and paraphrase candidate s.

$$\mathrm{AddCos}(s, C) = \frac{1}{|C|} \sum_{c \in C} \cos(\vec{s}, \vec{c}) \tag{2.12}$$

$$\mathrm{AvgCos}(s, C) = \cos(\vec{s}, \frac{1}{|C|} \sum_{c \in C} \vec{c}) \tag{2.13}$$

2.5. Evaluation for Japanese Lexical Simplification

In this section, we construct an evaluation dataset for Japanese lexical simplification. According to the previous works in English [99, 12], we also use crowdsourcing to construct the evaluation dataset for lexical simplification by the following procedure.

1. We prepare target sentences with a difficult word.

2. Annotators generate paraphrases of complex word in consideration of context.

3. Annotators rank complex word and its paraphrases in terms of simplicity.

4. We integrate rankings obtained from annotators.

In Section 2.5.1, we explain existing evaluation datasets for lexical simplification and summarize the improvements to build a better evaluation dataset. In Sections 2.5.2 to 2.5.5, we build an evaluation dataset for Japanese lexical simplification according to the above procedure using crowdsourcing[10]. In Section 2.5.6 introduce the evaluation metrics in lexical simplification task. For crowdsourcing, we requested the annotators to complete at least 95% of their previous assignments correctly. They were native Japanese speakers.

2.5.1 Previous Works

Four datasets have been constructed for English lexical simplification.

SemEval[11]

Specia et al. [99] reranked the dataset for English lexical substitution [71] to build the dataset for English lexical simplification. The data was selected from the English Internet Corpus[12] [95]. This is a balanced corpus similar in flavour to the BNC, though with less bias to British English, obtained by sampling data from the web. This dataset comprises 2,010 sentences, 201 target words each with 10 sentences. In paraphrase step, each target word was paraphrased by five native English speakers in consideration of context. In reranking step, each sentence was reranked by four or five non-native English speakers in terms of simplicity. In integration step, a gold-standard ranking was created based on the average rank of each word.

LSeval[13]

In common with Specia et al. [99], De Belder and Moens [12] reranked the dataset for English lexical substitution [71] to build the dataset for English lexical simplification. There are four major differences from Specia's work. (1)

[10] http://www.lancers.jp/

[11] https://www.cs.york.ac.uk/semeval-2012/task1/

[12] http://corpus.leeds.ac.uk/internet.html

[13] http://people.cs.kuleuven.be/~jan.debelder/lseval.zip

They excluded simple target words[14] from their dataset. (2) In their reranking step, they recruited five annotators for each sentence using crowdsourcing[15]. (3) In their reranking step, they allowed annotators to include tie ranks in the rankings. (4) In their integration step, they created a gold-standard ranking in consideration of reliability of each annotator using noisy channel model. This dataset comprises 430 sentences, 43 target words each with 10 sentences.

LexMTurk[16]

Horn et al. [44] directly acquired simple paraphrases using crowdsourcing[15]. Each 500 sentences from English Wikipedia have a target word, and 50 annotators for each sentence gave a simple paraphrase. Target words for simplification were selected from the words in English Wikipedia that changed in the parallel corpus of English Wikipedia and Simple English Wikipedia [27]. In their integration step, they created a gold-standard ranking by simply counting word frequency on annotations. That is, they defined that the paraphrase suggested by more annotators is a better simplification. This dataset comprises 500 sentences, 500 target words each with only one sentence.

BenchLS[17]

Paetzold and Specia [80] used a dataset consisting of 929 sentences that integrated LSeval [12] and LexMTurk [44] for benchmarking lexical simplification.

Based on the previous works above, we construct an evaluation dataset considering follows.

Target Selection

In order to build a better evaluation dataset, we carefully choose the target sentence and the target word. First, in order not to limit the diversity of expressions, the target sentence is selected from the balanced corpus. Next, we select complex words to be simplified for the target word. As pointed out by Specia et al. [99], it is natural for each word in the sentence to have consistent difficulty. In other words, we do not simplify only the target word in the complex context, but simplify the complex word that appear in the simple context. SemEval dataset selects target sentences from a balanced corpus, but includes simple words in

[14] http://simple.wikipedia.org/wiki/Wikipedia:Basic_English_combined_wordlist
[15] https://www.mturk.com/
[16] http://www.cs.pomona.edu/~dkauchak/simplification/
[17] https://gustavopaetzold.wordpress.com/resources/

Table 2.1: Evaluation Dataset for Lexical Simplification

	# Sents.	Avg. # Subs.	Target Selection	Tie Ranking	Exclude Outliers
SemEval	2,010	5.00	△	×	×
LSeval	430	5.04	△	✓	✓
LexMTurk	500	12.86	△	-	×
BenchLS	929	7.37	△	-	△
Ours	2,010	4.30	✓	✓	✓

target words. LSeval dataset selects target sentences from a balanced corpus and removes simple words from target words, but the context of the target word may be complex. In LexMTurk dataset, only target words are complex, but the source of the target sentence is limited to English Wikipedia. We select sentences with only one complex word from a balanced corpus as target sentences and choose the complex word as the target word for simplification.

Tie Ranking

In the reranking step, a tie cannot be assigned in SemEval dataset. This deteriorates ranking consistency if some substitutes have a similar simplicity. LSeval dataset allows ties in simplification ranking and De Belder and Moens report considerably higher agreement among annotators than SemEval dataset. We also allow tie to annotators in our reranking step. In LexMTurk dataset, since annotators only give simple paraphrases, there is no reranking step. However, we want to deal with the phenomenon "cannot be paraphrased into simple words", so we follow SemEval or LSeval datasets. There are at least one type of simple paraphrase because the parallel corpus is used as the source in the LexMTurk dataset. However, such a situation cannot be generally assumed.

Exclude Outliers

SemEval dataset uses an average score to integrate rankings, but it might be biased by outliers. LexMTurk dataset also treats all annotators equally. However, it is difficult to believe all annotators unconditionally in crowdsourcing. De Belder and Moens [12] report a slight increase in agreement by greedily removing annotators to maximize the agreement score. We propose better annotation integration method for crowdsourcing considering the reliability of each annotator.

2.5.2 Target Selection

We define complex words as "High Level" words in the JEV lexicon [105]. There were 7,939 complex words out of 17,920 words in the JEV lexicon. In addition, target words of this work comprised content words (nouns, verbs, adjectives, adverbs, adjectival nouns, sahen nouns[18], and sahen verbs[19]).

Sentences that include only one complex word were randomly extracted from the Balanced Corpus of Contemporary Written Japanese [67]. Sentences shorter than seven words or longer than 35 words were excluded. Replacing a word in a compound word can not hold the meaning of the compound word in many cases, so the target word appearing as a part of the compound word is excluded. Conjugation was allowed to cover variations of both verbs and adjectives. Following previous work [71, 99, 12], 10 contexts of occurrence were collected for each complex word. We assigned 30 complex words for each part of speech. The total number of sentences was 2,100 (30 words × 10 sentences × 7 parts of speech). We used a crowdsourcing to annotate 1,800 sentences, and we asked university students majoring in computer science to annotate 300 sentences.

2.5.3 Paraphrase Acquisition and Selection

For each complex word, five annotators gave as much paraphrase as possible without changing the meaning of sentence. Substitutions could include particles in context. An average of 4.59 paraphrases were given for 2,100 target words.

According to McCarthy and Navigli [71], we calculated pairwise agreement between each pair of sets $(a_1, a_2 \in A)$ from all the possible pairings P as inter-annotator agreement.

$$\mathrm{IAA}(A) = \frac{1}{|A|} \sum_{a_1, a_2 \in A} \frac{a_1 \cap a_2}{a_1 \cup a_2} \tag{2.14}$$

The IAA for our paraphrase acquisition step was 0.194, which was a low score. This is because each annotator gave as much paraphrase as possible to acquire various paraphrases, because there were mixed annotators that give many paraphrases and annotators that give only a few paraphrases.

[18] Sahen noun is a kind of noun that can form a verb by adding a generic verb "する (do)" to the noun. (e.g. "修理 (repair)")

[19] Sahen verb is a sahen noun that accompanies with "する (do)". (e.g. "修理する (do repair)")

To improve the quality of the lexical substitution, inappropriate substitutes were deleted for later use. Another five annotators selected an appropriate word to include as a substitution that did not change the sense of the sentence. The IAA for our paraphrases selection step was 0.669, which was greatly improved.

Substitutes that won a majority were defined as correct. Nine complex words that were evaluated as not having substitutes were excluded at this point. As a result, an average of 4.30 paraphrases were given for 2,010 target words.

2.5.4 Simplicity Reranking

Another five annotators arranged substitutes and complex words according to the simplification ranking. Annotators were permitted to assign a tie, but they could select up to four items[20] to be in a tie because we intended to prohibit an insincere person from selecting a tie for all items.

According to De Belder and Moens [12], we calculated Spearman rank correlation coefficient as inter-annotator agreement.

$$\rho(i,j) = \frac{\sum_k (\mathrm{rank}_i(w_k) - \overline{\mathrm{rank}_i})(\mathrm{rank}_j(w_k) - \overline{\mathrm{rank}_j})}{\sqrt{\sum_k (\mathrm{rank}_i(w_k) - \overline{\mathrm{rank}_i})^2 \sum_k (\mathrm{rank}_j(w_k) - \overline{\mathrm{rank}_j})^2}} \tag{2.15}$$

Here, let me define w_k the k-th word in the list of substitutions, $\mathrm{rank}_i(w_k)$ the rank given by annotator i to word w_k, and $\overline{\mathrm{rank}_i}$ the average rank of the words given by annotator i. Often words are ranked at the same position by the annotators, and ties here are solved by assigning them the average of their rank. The Spearman's ρ for our simplicity reranking step was 0.552, which was moderate correlation.

2.5.5 Integrating Annotations

Annotators' rankings were integrated into one ranking, using a maximum likelihood estimation [70] to penalize deceptive annotators.

$$\begin{aligned} \Pr[\tilde{\pi} \mid \pi, \lambda^{(i)}] &= \frac{1}{Z(\lambda^{(i)})} \exp\left(-\lambda^{(i)} d(\tilde{\pi}, \pi)\right) \\ Z(\lambda^{(i)}) &= \sum_{\tilde{\pi}} \exp\left(-\lambda^{(i)} d(\tilde{\pi}, \pi)\right) \end{aligned} \tag{2.16}$$

[20] In annotations by university students who can expect honesty, up to four tie ranks have appeared.

This model gives the probability of a rank vector $\tilde{\pi}$, given a modal order π and a i-th annotator's concentration parameter $\lambda^{(i)}$. Here, $d(\cdot,\cdot)$ denotes a distance between two rank vectors, and $Z(\lambda^{(i)})$ is a normalizing constant. We employ the Spearman rank correlation coefficient as a distance. In this model, the annotation of an annotator who has a high concentration parameter $\lambda^{(i)}$ is likely to be an accurate order whose distance from the true order is small. Therefore, this method estimates the reliability of annotators in addition to determining the true order of rankings. We applied the reliability score to exclude extraordinary annotators.

We excluded annotators with low reliability score under the constraint of excluding only up to two annotators out of five for each target sentence. As a result, nine annotators out of 140 were excluded and the Spearman's ρ between gold-standard and each annotator was improved from 0.541 to 0.580. After excluding annotators with low reliability, annotations were integrated into gold-standard ranking using the average ranking according to the SemEval dataset.

2.5.6 Metrics

Lexical simplification methods are automatically evaluated by following three metrics using gold-standard ranking.

Precision: The proportion of instances in which the highest ranking substituion is either the target complex word itself or is in the gold-standard.

Accuracy: The proportion of instances in which the highest ranking substituion is not the target complex word itself and is in the gold-standard.

Changed Proportion: The proportion of instances in which the highest ranking substituion is not the target complex word itself.

The most important metric is Accuracy. Because Precision gives a high score to a safe system that rarely rewrites. Moreover, Changed Proportion gives a high score to a harmful system that frequently rewrites.

2.6. Experiments

In this section, the proposed methods described in Sectons 2.1 to 2.4 are evaluated by Precision, Accuracy and Changed Proportion according to Section 2.5.

Table 2.2: Paraphrase Lexicons

	Vocab	Paraphrases
Iwanami	38,100	325,636
Psylex	23,614	170,235
Manual	165,318	253,092
Automatic	120,576	842,492

2.6.1 Settings

In our proposed methods, simplification candidates are extracted from three sources: pairs of headword and definition statement, synonym dictionaries constructed manually, and synonym dictionaries automatically constructed from bilingual parallel corpora. Annotated Corpus of Iwanami Japanese Dictionary Fifth Edition 2004[21] (Iwanami) and Basic Word Database[22] (Psylex) are used as pairs of headword and definition statement. Moreover, we combine the five synonym dictionaries (Lexical Paraphrase Dictionary of Japanese Content Words, Japanese WordNet Synonyms Database, Verb Entailment Database, Database of Japanese Orthographic Variant Pairs, and Case Base for Basic Semantic Relations) described in Section 2.1.2 as a synonym dictionary constructed manually (Manual). As the automatically constructed synonym dictionary (Automatic), we use the largest version ("10best") of PPDB: Japanese[23] [76]. PPDB: Japanese consists of a paraphrase pair of up to 7-gram acquired from 1.9 million pairs of Japanese-English parallel semtemces using bilingual pivoting [10] described in Sections 2.1.3 and 2.2.3. In the lexicon, phrases with top 10 paraphrase probabilities for each phrase are included. We use only 1-gram pairs. Table 2.2 shows the vocabulary size and the number of paraphrase rules for each Paraphrase lexicon.

For meaning preservation filtering, we use Japanese WordNet[2] (Ver.1.1) for PDS. We use CBOW model [73] of the 200-dimensional word2vec embeddings[24] for WES. The word embeddings were trained on 12 million sentences of Japanese Wikipedia[25] split into words using MeCab [57] (Ver. 0.996) with IPADIC (Ver. 2.7.0).

For simplicity filtering, we use Basic Word Database[22] for calculating word familiarity. The word frequency was calculated on Japanese Wikipedia[25].

[21] http://www.gsk.or.jp/catalog/gsk2010-a/
[22] http://hon.gakken.jp/book/1530238600
[23] http://ahclab.naist.jp/resource/jppdb/
[24] https://code.google.com/archive/p/word2vec/
[25] https://dumps.wikimedia.org/jawiki/20161001/

For grammaticality ranking, we trained a 5-gram language model from Japanese Wikipedia[25] using KenLM [42] for LM.

2.6.2 Baseline: LIGHT-LS

We compare the proposed method with LIGHT-LS [34] as a Baseline which is a lexical simplification method without simplified corpora. Glavaš and Štajner performs lexical simplification based on a monolingual corpus using following two steps.

1. **Candidate Selection:** The 10 most similar candidate words are selected for each difficult word using cosine similarity between word embeddings.

2. **Reranking:** The best candidate is selected using the average ranking based on the following four features.

   Semantic Similarity They select candidates with high cosine similarity cos between the word embeddings $\vec{t}$ of the target word t and the word embeddings $\vec{s}$ of the simplification candidate s. This feature corresponds to the our meaning preservation filtering.

   $$f_1(t,s) = \cos(\vec{t},\vec{s}) \tag{2.17}$$

   Context Similarity They select candidates with high averaged cosine similarity cos between the word embeddings $\vec{s}$ of the simplification candidate s and the word embeddings $\vec{c}$ of each context word $c \in C$. Here, C is a set of content words (nouns, verbs, adjectives, and adverbs) in the target sentence. This feature corresponds to the our grammaticality ranking.

   $$f_2(C,s) = \frac{1}{|C|}\sum_{c\in C}\cos(\vec{c},\vec{s}) \tag{2.18}$$

   Information Contents Under the hypothesis that the word's informativeness correlates with its complexity [29], they select less informative candidates. Here, W is a vocabulary, and freq is a word frequency. This feature corresponds to our simplicity filtering.

   $$f_3(W,s) = -\log\frac{\text{freq}(s)+1}{\sum_{w\in W}\text{freq}(w)+1} \tag{2.19}$$

Table 2.3: Japanese Lexical Simplification

	M	S	G	Precision	Changed	Accuracy	Oracle
Baseline	LIGHT-LS			0.533	0.544	0.077	0.160
Iwanami	WES	None	LM	0.810	0.236	0.046	0.093
Psylex	WES	None	LM	0.649	0.427	0.076	0.167
Manual	None	None	LM	0.805	0.309	0.114	0.168
Automatic	MIPA	None	LM	0.668	0.430	0.098	0.211

Language Model They select candidates with high probability of N-gram language model ($N = 5$). This feature corresponds to our grammaticality ranking.

$$f_4(w_1, w_2, ..., w_n) = \prod_{k=1}^{n} p(w_k \mid w_{k-N+1}^{k-1}) \tag{2.20}$$

2.6.3 Results

Table 2.3 shows the experimental results on Japanese lexical simplification. In the method using definition statements (Psylex), using WES as the meaning preservation filtering and LM as grammaticality ranking without simplicity filtering, we achieved the same performance as LIGHT-LS which is a state-of-the-art lexical simplification method without simplified corpora. In the method using synonym dictionaries constructed manually, we achieved the best performance on Japanese lexical simplification task using LM as a grammaticality ranking without filtering methods. In the method using synonym dictionary constructed automatically, we outperformed LIGHT-LS baseline using MIPA as meaning preservation filtering and LM as grammaticality ranking without simplicity filtering.

In terms of Oracle Accuracy selecting the best candidate, our Psylex method and Manual method achieved the same performance as LIGHT-LS baseline. Since our Manual method can use high-quality paraphrase lexicon, the method achieved the best Precision. However, since our Automatic method can use large-scale paraphrase lexicon, the method achieved the best Oracle Accuracy. Future works include maximizing the use of automatically constructed paraphrase lexicon by extending to phrases and improving both filtering and ranking method.

Chapter 3

Sentence Simplification without Simplified Corpora

In this chapter, we assume a language that cannot use a large-scale simplified corpora, construct a pseudo-parallel corpus for text simplification from a raw corpus, and perform text simplification using a statistical machine translation. We use readability assessment method and sentence alignment method to search simplified synonymous sentences for each complex sentence in a given monolingual corpus. Using the sentence pairs, the PBSMT model acquires phrase pairs to translate complex expressions into simpler synonymous expressions.

First, Section 3.1 outlines the proposed method of constructing a pseudo-parallel corpus from a raw corpus. Next, Section 3.2 proposes an sentence similarity estimation method based on alignment between word embeddings as sentence alignment for text simplification. Moreover, experiments are presented in Sections 3.3 to 3.5. First, Section 3.3 evaluates the proposed method from Section 3.2 and determines the best sentence alignment method for text simplification. Section 3.4 constructs an English pseudo-parallel corpus based on Sections 3.1 to 3.3, and performs English text simplification. Section 3.5 similarly builds a Japanese pseudo-parallel corpus and performs Japanese text simplification.

3.1. Pseudo-Parallel Corpus from a Raw Corpus

Recent studies have treated text simplification as a monolingual machine translation problem wherein a simple synonymous sentence is generated using phrase-based

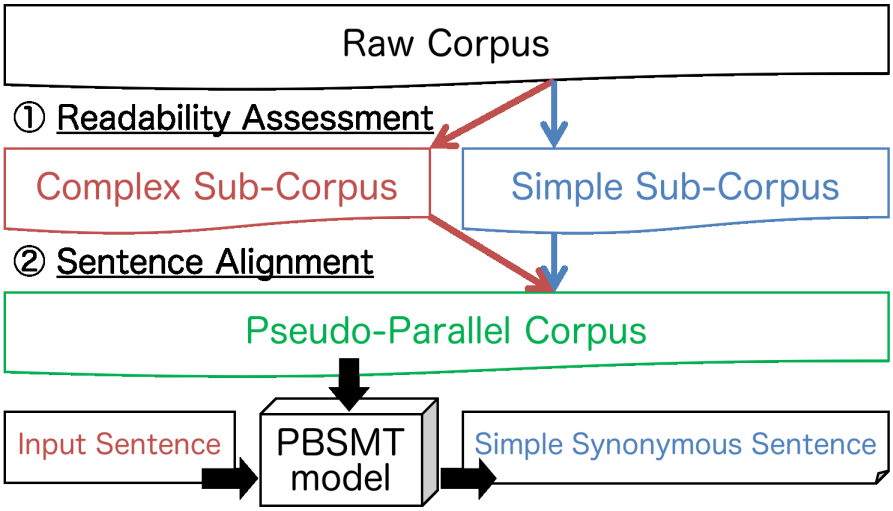

Figure 3.1: Text simplification using PBSMT from only a raw corpus by readability assessment and sentence alignment.

statistical machine translation (PBSMT) [98, 27, 26, 120, 111, 118, 113, 112, 35]. However, building a monolingual parallel corpus for text simplification is costly because a large-scale corpus written in simple expressions is not publicly available in many languages other than English. Hence, text simplification was studied mainly in English for where rich resources are available such as a manually constructed text simplification corpus [121], a large-scale simplified corpus (Simple English Wikipedia[1]), and a paraphrase database [33, 86, 85].

Therefore, we propose a language-independent unsupervised method that automatically builds a pseudo-parallel corpus to train a text simplification model from only a raw corpus. Synonymous or similar sentence pairs, such as multiple mentions or explanations of similar events or items, could be obtained from a large-scale monolingual corpus. We carefully create a parallel corpus containing complex form on one part and simple form on the other part. We automatically acquire such sentence pairs from the raw corpus. Our novel framework comprises two steps: 1) readability assessment and 2) sentence alignment. An overview of the proposed method is shown in Figure 3.1.

In this research, we propose a framework for automatically constructing a pseudo-parallel corpus for text simplification from a raw corpus. This can be explained more generally as in Figure 3.2. In other words, for randomly extract two sentences from

[1] http://simple.wikipedia.org/

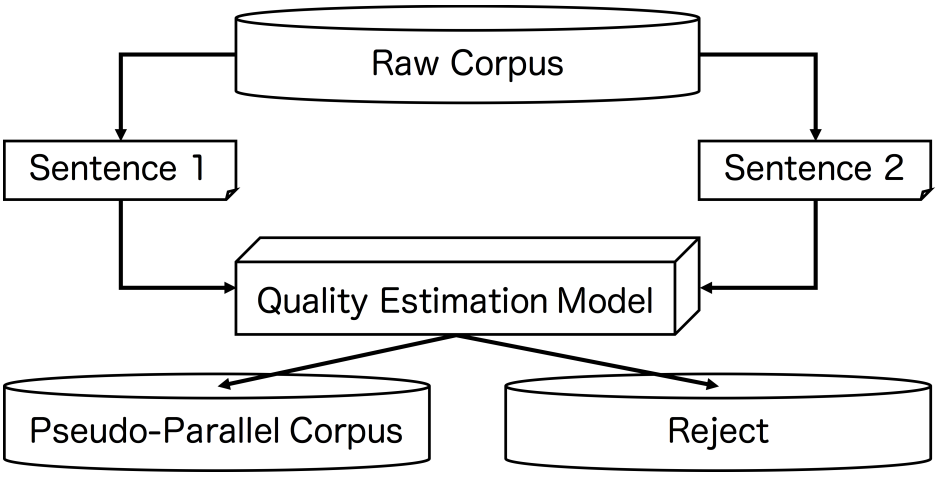

Figure 3.2: Pseudo-parallel corpus from a raw corpus.

the raw corpus, we perform a quality estimation according to the task, and extract sentence pairs with likelihood above the threshold as a pseudo-parallel corpus. Quality estimation [102] is a generic term for technologies to evaluate output sentences without reference by comparing input and output sentences, and is studied mainly in Text-to-Text generation tasks, especially in machine translation [23, 17, 18, 21, 20, 19].

We would like to build a pseudo-parallel corpus for text simplification. Since text simplification is a task that rewrites from complex sentence into simpler version while preserving its meaning, the quality estimation step in the Figure 3.2 evaluates the difficulty of each sentence and the synonymity between two sentences. In order to evaluate difficulty of sentence, we use the readability metrics developed in each language. After estimating readability for each sentence, we next evaluate the synonymity of complex sentences with low readability and simple sentences with high readability. In general, it is easier to read short sentences than long sentences, so in addition to paraphrasing from complex expressions into simple ones, text simplification often deletes expressions that are not important [121]. Hence, synonymy in text simplification is not limited to mutually replaceable "synonymity" as in paraphrase tasks. Therefore, we evaluate the synonymity between two sentences using the sentence similarity described in Section 3.2. Finally, only pairs of complex and simple sentences which have high similarity are used as a pseudo-parallel corpus for text simplification.

There are previous works to construct a pseudo-parallel corpus from a raw corpus such as Suzuki et al. [107], Sennrich et al. [93], Imankulova et al. [47]. In order to construct a pseudo-parallel corpus for paraphrase identification, Suzuki et al. translated

sentence extracted from the raw corpus using two kinds of machine translation systems, generated two types of translated sentences, A and B, and estimated the quality of the translated pair (A, B). Sennrich et al. and Imankulova et al. extracted sentence A from the raw corpus and translated it into sentence B and constructed a pseudo-parallel corpus from a translated pair (A, B). Here, Imankulova et al. used quality estimation, but Sennrich et al. did not use them. In these previous works, since sentences generated by machine translation systems are used as pseudo-parallel corpus, erroneous sentences may be included due to translation errors. On the other hand, in this work, since sentences extracted from the raw corpus are used, erroneous sentences are not included. The pseudo-parallel corpus constructed using our approach was also reported usefulness in domain adaptation of machine translation [68].

3.2. Sentence Alignment Based on Alignment between Word Embeddings

Three monolingual parallel corpora for English text simplification have been built from English Wikipedia and Simple English Wikipedia. First, Zhu et al.[2] [128] pioneered automatic construction of a text simplification corpus using the cosine similarity between sentences represented as TF-IDF vectors. Second, Coster and Kauchak[3] [27] extended Zhu et al.'s work by considering the order of the sentences. However, these methods did not compute similarities between different words. In text simplification, it would be useful to consider similarities between synonymous expressions when computing the similarity between sentences, since concepts are frequently rewritten from a complex to a simpler form. Third, Hwang et al.[4] [46] computed the similarity between sentences taking account of wordlevel similarity using the co-occurrence of a headword in a dictionary and its definition sentence. We also consider word-level similarity to compute similarity between sentences but using word embeddings to build a text simplification corpus at low cost without requiring access to external resources.

To address the challenge of computing the similarity between sentences containing different words with similar meanings, many methods have been proposed. In semantic textual similarity task [5, 6, 3, 2, 4, 24], sentence similarity is computed on the basis

[2]https://www.ukp.tu-darmstadt.de/data/sentence-simplification/simple-complex-sentence-pairs/
[3]http://www.cs.pomona.edu/~dkauchak/simplification/
[4]http://ssli.ee.washington.edu/tial/projects/simplification/

of word similarity following the success of word embeddings [73]. For example, a supervised approach using word embeddings when obtaining a word alignment achieved the best performance in SemEval-2015 Task 2 [104]. Word embeddings have also been used in unsupervised sentence similarity metrics [74, 97, 60]. These unsupervised sentence similarity metrics can be applied to the automatic construction of a monolingual parallel corpus for text simplification, without requiring the data to be labeled.

We propose four types of sentence similarity measures for building a monolingual parallel corpus for text simplification, based on alignments between word embeddings that have achieved outstanding performance on different NLP tasks. AAS, MAS, HAS are the sentence similarity measures proposed by Song and Roth [97] for a short text similarity task. The Word Mover's Distance (WMD) [60] is another sentence similarity measure based on alignment between word embeddings that is known to achieve good performance on a document classification task.

3.2.1 AAS: Average Alignment Similarity

AAS [97] averages the cosine similarities between all pairs of words within given two sentences, x and y, calculated over their embeddings.

$$\mathrm{AAS}(x,y) = \frac{1}{|x||y|}\sum_{i=1}^{|x|}\sum_{j=1}^{|y|}\cos(\vec{x}_i,\vec{y}_j) \tag{3.1}$$

3.2.2 MAS: Maximum Alignment Similarity

AAS inevitably involves noise, as many word pairs are semantically irrelevant to each other. MAS [97] reduces this kind of noise by considering only the best word alignment for each word in one sentence as follows.

$$\mathrm{MAS}_{\mathrm{asym}}(x,y) = \frac{1}{|x|}\sum_{i=1}^{|x|}\max_{j}\cos(\vec{x}_i,\vec{y}_j) \tag{3.2}$$

Here, MAS is an inherently asymmetric score. Therefore, we obtain the symmetric sentence similarity by averaging each direction as follows.

$$\mathrm{MAS}(x,y) = \frac{1}{2}\mathrm{MAS}_{\mathrm{asym}}(x,y) + \frac{1}{2}\mathrm{MAS}_{\mathrm{asym}}(y,x) \tag{3.3}$$

3.2.3 HAS: Hungarian Alignment Similarity

AAS and MAS deal with many-to-many and one-to-many word alignments, respectively. On the other hand, HAS [97] is based on one-to-one word alignments.

The task of identifying the best one-to-one word alignments $\mathscr{H}$ is regarded as a problem of bipartite graph matching, where the two sets of vertices respectively comprise words within each sentence x and y, and the weight of a edge between x_i and y_j is given by the cosine similarity calculated over their word embeddings. Given $\mathscr{H}$ identified using the Hungarian algorithm [58], HAS is computed by averaging the cosine similarities between embeddings of the aligned pairs of words.

$$\mathrm{HAS}(x,y) = \frac{1}{|\mathscr{H}|} \sum_{(i,j)\in\mathscr{H}} \cos(\vec{x}_i, \vec{y}_j) \tag{3.4}$$

where $|\mathscr{H}| = \min(|x|,|y|)$, as $\mathscr{H}$ contains only one-to-one word alignments.

3.2.4 WMD: Word Mover's Distance

WMD [60] is a special case of the Earth Mover's Distance [92], which solves the transportation problem of words between two sentences represented by a bipartite graph.[5] Let n be the vocabulary size of the language, WMD is computed as follows.

$$\mathrm{WMD}(x,y) = \min \sum_{u=1}^{n} \sum_{v=1}^{n} \mathscr{A}_{uv} \mathrm{eud}(\vec{x}_u, \vec{y}_v) \tag{3.5}$$

$$\text{subject to : } \sum_{v=1}^{n} \mathscr{A}_{uv} = \frac{1}{|x|} freq(x_u, x)$$

$$\sum_{u=1}^{n} \mathscr{A}_{uv} = \frac{1}{|y|} freq(y_v, y)$$

where $\mathscr{A}_{uv}$ is a nonnegative weight matrix representing the flow from a word x_u in x to a word y_v in y, $\mathrm{eud}(\cdot,\cdot)$ the Euclidean distance between two word embeddings, and $freq(\cdot,\cdot)$ the frequency of a word in a sentence.

[5] Note that the vertices in the graph represent the word types, unlike the token-based graph for HAS.

3.3. Experiment: Alignment within Complex and Simple Sentences

In this section, we perform parallel and nonparallel binary classification on pairs of complex and simple sentences, and evaluate the effectiveness of sentence similarity based on the alignment between word embeddings.

3.3.1 Settings

Hwang et al. [46] built a benchmark dataset[4] for text simplification extracted from the English Wikipedia and Simple English Wikipedia. They annotated one of the following four labels to 67,853 sentence pairs:

Good (G): The semantics of the sentences completely match, possibly with small omissions. 277 sentence pairs.

Good Partial (GP): A sentence completely covers the other sentence, but contains an additional clause or phrase that has information which is not contained within the other sentence. 281 sentence pairs.

Partial (P): The sentences discuss unrelated concepts, but share a short related phrase that does not match considerably. 117 sentence pairs.

Bad (B): The sentences discuss unrelated concepts. 67,178 sentence pairs.

We classified a sentence pair as parallel or nonparallel using this dataset to evaluate the sentence similarity measures. We conducted experiments in two settings:

G vs. O: Only sentence pairs labeled G were defined as parallel, and others (O) were defined as nonparallel.

G+GP vs. O: Sentence pairs labeled either G or GP were defined as parallel.

We evaluated the performance of the binary classification using following two measures in accordance with Hwang et al. [46]:

MaxF1: The maximum F1 score.

AUC-PR: Area under the curve on the precision-recall curve.

Table 3.1: Binary classification accuracy of parallel and nonparallel sentences.

		G vs. O		G+GP vs. O	
		MaxF1	AUC-PR	MaxF1	AUC-PR
Baseline: Zhu	(Hwang et al., 2015)	0.550	0.509	0.431	0.391
Baseline: Coster	(Hwang et al., 2015)	0.564	0.495	0.415	0.387
Baseline: Hwang	(Hwang et al., 2015)	0.712	0.694	0.607	0.529
Baseline: Additive Embeddings Similarity		0.691	0.695	0.518	0.487
AAS: Average Alignment Similarity		0.419	0.312	0.391	0.297
MAS: Maximum Alignment Similarity		0.717	0.730	**0.638**	**0.618**
HAS: Hungarian Alignment Similarity		0.524	0.414	0.354	0.275
WMD: Word Mover's Distance		**0.724**	**0.738**	0.531	0.499

Noise in the word alignment for AAS, MAS, and HAS was removed by aligning only those word pairs (x_i, y_j) which had a word similarity $\cos(x_i, y_j) > \theta$. This threshold θ was tuned to maximize MaxF1. We employed 0.89 and 0.95 in the binary classification of G vs. O and G+GP vs. O for AAS, 0.28 and 0.49 in the binary classification of G vs. O and G+GP vs. O for MAS, and 0.98 in the binary classification of G vs. O and G+GP vs. O for HAS.

Table 3.1 compares sentence similarity measures in the binary parallel and nonparallel classification task. The top three methods in the upper row are taken from previous studies of monolingual parallel corpus construction for text simplification [128, 27, 46], and the five methods in the lower rows are the sentence similarity measures based on the word embeddings. Additive embeddings provides yet another baseline method, in which sentence embeddings are composed by adding word embeddings without word alignment, and sentence similarity is computed using the cosine similarity between sentence embeddings. We used publicly available[6] pretrained word embeddings to compute sentence similarity.

3.3.2 Results

From Table 3.1, it can be seen that WMD performed best in the binary classification task between G vs. O, whereas MAS performed best in the binary classification task between G+GP vs. O.

[6]https://code.google.com/archive/p/word2vec/

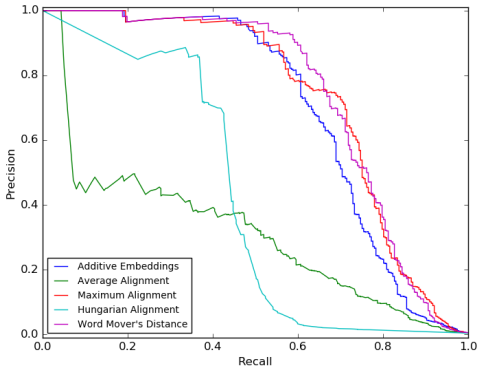

Figure 3.3: PR curves in binary classification of G vs. O.

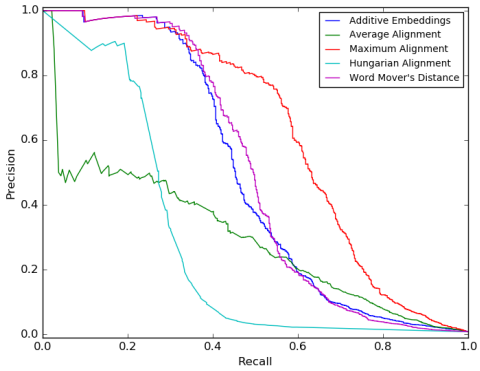

Figure 3.4: PR curves in binary classification of G+GP vs. O.

Figures 3.3 and 3.4 show the Precision-Recall curves in the binary classification task between parallel and nonparallel sentences. Figure 3.4 shows that MAS performed better than the other sentence similarity measures based on word embeddings, in the binary classification between G+GP vs. O.

Text simplification must take account not only of paraphrases from a complex expression to a simple expression but also of the deletion of unimportant parts of a complex sentence. It is therefore important to include both G sentence pairs, where the simple sentence is synonymous with the complex sentence, and GP sentence pairs, where the complex sentence entails the simple sentence. For this reason, MAS, which performed best in classification between G+GP vs. O, was the preferred measure for computing sentence similarity in text simplification.

3.4. Experiment: English Sentence Simplification

We automatically construct a pseudo-parallel corpus for text simplification from a raw corpus by two steps combining readability assessment and sentence alignment, as shown in Figure 3.1. We first calculate the readability of sentences and divide a raw corpus into two sub-corpora comprising complex and simple sentences. Then, we compute sentence similarity using word embeddings for all pairs of complex and simple sentences, and build a monolingual pseudo-parallel corpus to train a text simplification model by extracting sentence pairs with high sentence similarity. By training a PBSMT model using such a corpus, it is possible to generate simple synonymous sentences from input sentences.

3.4.1 English Pseudo-Parallel Corpus for Text Simplification

We use English Wikipedia as a raw corpus and Flesch Reading Ease [32] as a readability measure. Flesch Reading Ease is well known for English readability assessment and is often used for English text simplification [128, 15].

$$\text{FRE} = 206.835 - 1.015 \times (\text{\# of words}) - 84.6 \times (\text{Avg. \# of syllables}) \tag{3.6}$$

The FRE score ranges from 0 to 100 wherein $[60, 70)$ as a standard; the higher the score, the more readable it is. Thus, we divide English Wikipedia[7] into a complex corpus comprising sentences with a readability $[0, 60)$ and a simple corpus comprising the remaining sentences. This results in 3,689,227 sentences for the complex corpus and 2,358,921 sentences for the simple corpus.

We use publicly available pretrained word embeddings[6] for alignment by MAS. In order to reduce noise, only word pairs with a word similarity $[0.5, 1.0]$ are used for word alignment. As a result, 2,072,572 sentence pairs with a sentence similarity of $[0.5, 1.0)$[8] are extracted greedily into our pseudo-parallel corpus.

Figure 3.5 shows the distribution of the readability of the sentences of English Wikipedia and Simple English Wikipedia. The vertical axis is the normalized frequency of sentences for each readability score. In the range of less than 60 of the readability based on the FRE score, the sentences from English Wikipedia which is a complex corpus appears at a high rate. Similarly, in the range of 60 or more, the sentences from Simple English Wikipedia which is a simplified corpus appears at a high rate. Therefore, we can conclude that the threshold of 60, which divides complex and simple sentences, is valid. Moreover, while English Wikipedia has a lot of complex sentences, it also shows that not all sentences are complex. Hence, by extracting simpler sentences from English Wikipedia, it is possible to obtain a simple sub-corpus without relying on simplified corpora.

Figure 3.6 shows the quality of sentence pairs for each similarity range. Two annotators gave labels[9] (Good, Good Partial, Partial, Bad) to 500 sentence pairs following Hwang et al. [46]. The higher the sentence similarity is, the less "Bad" sentence pairs are.

Table 3.2 shows examples from our text simplification corpus. In the sentence pair of "Good", there are shown examples of paraphrase (precipitation $\rightarrow$ rainfall) from

[7] https://dumps.wikimedia.org/enwiki/20160501/

[8] We adopt sentence similarity $[0.5, 1.0)$ following Zhu et al. [128], but if any evaluation corpus is available the threshold can be optimized.

[9] Pearson correlation coefficient reaches 0.629.

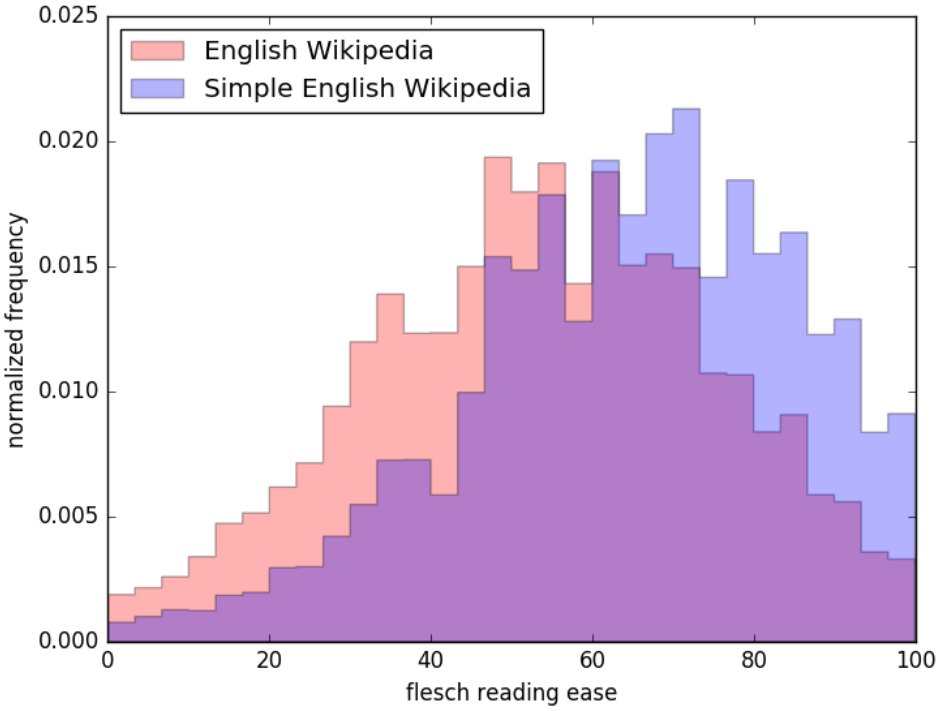

Figure 3.5: Readability score distribution of English Wikipedia and Simple English Wikipedia. A higher score in Flesch Reading Ease indicates simpler sentences.

complex word to simple one. An example of deletion can be seen in the sentence pair of "Good Partial". The sentence pair of "Partial" is not in a synonymous or entailment relation, but it contains common phrases and related phrases.

Unlike the pair of English Wikipedia and Simple English Wikipedia, the pair of complex sub-corpus and simple sub-corpus which divided English Wikipedia is not a comparable corpus. Therefore, as shown in Figure 3.6, the proportion of sentence pairs that are synonymous or entailment is not high. However, in this work, since text simplification is performed using phrase-based statistical machine translation, the influence of this problem is small, and important simplification rules can be acquired even from noisy sentence pairs for the following three reasons.

- Since text simplification is a problem of monolingual translation, it is possible to output many words in the input sentence as is (it is correct not to convert). Therefore, unlike a problem of bilingual translation, it is not a serious problem

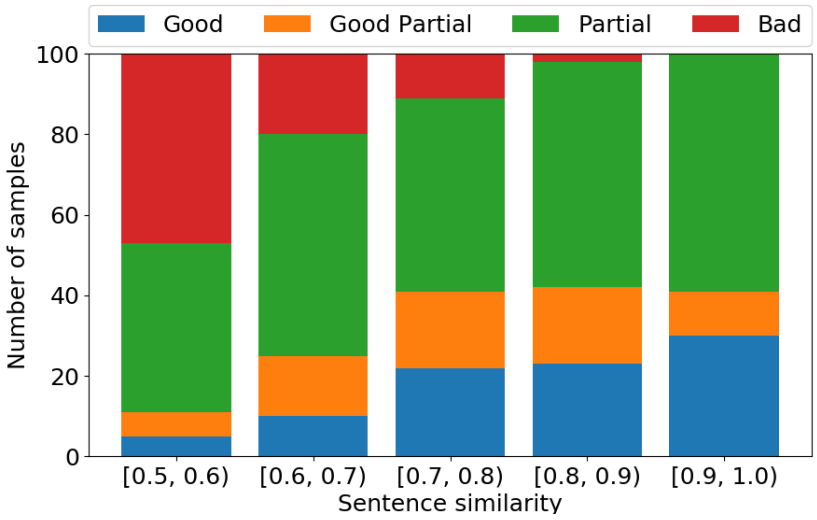

Figure 3.6: Quality of the pseudo-parallel corpus.

that only a small amount of appropriate phrase pairs are acquired.

- Phrase-based statistical machine translation learns to pairs in phrase level. Pairs of complex phrase and simpler paraphrase can be obtained not only from sentence pairs in synonymous or entailment relation but also from a pair of similar sentences.
- Phrase-based statistical machine translation finally reranks using language model, so if the appropriate phrase pair is included, simpler synonymous sentences can be obtained even if a lot of noisy phrase pairs are acquired.

3.4.2 Settings

We trained text simplification models using our pseudo-parallel corpus and existing text simplification corpora. The results were compared to evaluate the effectiveness of our text simplification corpus. We treated text simplification as a translation problem from the complex sentence to the simple one and modeled it using a phrase-based SMT trained as a log linear model.

Table 3.2: Examples of each label from our pseudo-parallel corpus. Good: synonymous sentence pair, Good Partial: a sentence completely covers the other sentence, Partial: sentence pair shares a short related phrase.

Label	Complex Sentence	Simple Sentence
Good	Climate in this area has mild differences between highs and lows, and there is adequate **precipitation** year round.	Climate in this area has mild differences between highs and lows, and there is adequate **rainfall** year round.
Good Partial	The new German Empire included 25 states (three of them, Hanseatic cities) **and the imperial territory of Alsace-Lorraine**.	The new German Empire included 25 states, three of them Hanseatic cities.
Partial	In 1996, she received the Primetime Emmy Award for Outstanding Supporting Actress in a Comedy Series, an award she was nominated for on seven occasions.	In 2006 and 2008, she received Emmy nominations for Outstanding Supporting Actress in a Drama Series.

$$\begin{aligned}\hat{s} &= \underset{\text{simple}}{\operatorname{argmax}}\, p(\text{simple} \mid \text{complex}) \\ &= \underset{\text{simple}}{\operatorname{argmax}}\, p(\text{complex} \mid \text{simple}) p(\text{simple}) \\ &= \underset{\text{simple}}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m h_m(\text{complex}, \text{simple})\end{aligned} \tag{3.7}$$

The log-linear model considers M feature functions $h_m(\text{complex}, \text{simple})$ and the weights of each feature λ_m, and models the translation probability $p(\text{simple} \mid \text{complex})$. In text simplification, we consider the searching problem for a simple sentence $\hat{s}$ which maximizes the weighted linear sum of feature functions for complex input sentences. As a feature function, we use the phrase simplification model $\log p(\text{complex} \mid \text{simple})$ and the language model $\log p(\text{simple})$ etc.

We used Moses 2.1 [56] as the PBSMT tool, GIZA++ [79] to obtain the word alignment, and KenLM [42] to build the 5-gram language model from the simple side of each text simplification corpus. For evaluation, we used a multiple reference

Table 3.3: Statistics of text simplification corpora.

	# sents.	# vocab complex	# vocab simple	length complex	length simple
Zhu et al. corpus	108,016	181,459	149,643	21.2	17.4
Coster and Kauchak corpus	137,362	132,567	120,620	23.6	21.1
Hwang et al. corpus	284,738	212,138	164,979	26.0	19.8
Our parallel corpus	492,993	274,775	198,043	25.3	17.9
Our pseudo-parallel corpus	2,072,572	174,310	156,271	43.5	32.7

dataset[10] [122] in which eight annotators have given simple synonymous sentences to 350 sentences extracted from English Wikipedia[11]. We automatically evaluated by FRE [32], BLEU [84] and SARI [122].

Table 3.3 shows statistics of text simplification corpora. Zhu et al. corpus [128], Coster and Kauchak corpus [27], and Hwang et al. corpus [46] are text simplification corpora built from English Wikipedia and Simple English Wikipedia. Our parallel corpus[12] is also a text simplification corpus built from English Wikipedia and Simple English Wikipedia but using MAS sentence alignment. Our pseudo-parallel corpus is a text simplification corpus built from only English Wikipedia.

Our parallel corpus gave a larger difference in the average number of words between complex and simple sentences than the other corpora, with values closer to the average numbers of words per sentence in the entire Wikipedia (25.1 and 16.9, respectively). This suggests that MAS was able to compute sentence similarity more accurately than the other measures regardless of the sentence length.

3.4.3 Results

Table 3.4 shows the text simplification performance. Baseline does not do any simplification. BLEU evaluates the meaning preservation and grammaticality such that the baseline that does not change any input sentence has the highest score. SARI also evaluates simplicity. Surprisingly, even without the help of simplified corpus, SARI reached the same level as the others using a large-scale simplified corpus, and BLEU also remains.

[10] https://github.com/cocoxu/simplification/

[11] We excluded English Wikipedia sentences included in test data from training data.

[12] https://github.com/tmu-nlp/sscorpus

Table 3.4: Results of English text simplification.

	# sents.	# rules	FRE	BLEU	SARI
Baseline	0	0	54.5	99.4	25.9
Zhu et al. corpus	108,016	7,441,535	59.7	84.7	34.7
Coster and Kauchak corpus	137,362	11,871,929	59.8	86.4	34.1
Hwang et al. corpus	284,738	25,482,261	61.0	81.3	34.5
Our parallel corpus	492,993	34,370,284	61.7	78.4	34.9
Our pseudo-parallel corpus	2,072,572	146,522,360	58.9	78.0	34.0

Table 3.5: Performance on each our pseudo-parallel corpus size.

Threshold for MAS	# sents.	# rules	FRE	BLEU	SARI
MAS $\geq$ 0.94	100,000	2,443,146	54.9	94.9	29.1
MAS $\geq$ 0.79	500,000	10,888,446	55.3	92.7	31.1
MAS $\geq$ 0.64	1,000,000	32,368,746	56.9	88.0	33.7
MAS $\geq$ 0.55	1,500,000	77,426,785	58.2	83.2	34.4
MAS $\geq$ 0.51	2,000,000	138,102,965	59.2	79.1	34.1
MAS $\geq$ 0.50	2,072,572	146,522,360	58.9	78.0	34.0

Table 3.5 shows the performance on each sentence similarity threshold. Sentence pairs with high similarity are less noisy, but include few simplification rules because the edit distance between complex and simple sentences is small. Due to the trade-off between the amount of simplification rules and noise contained in the pseudo-parallel corpus, the model trained by the corpus of 1.5M sentence pairs archived the highest SARI.

Table 3.6 shows examples of simplification trained with each text simplification corpus. Despite the fact that we do not use simplified corpora, we generated simple sentence similar to references using a large-scale simplified corpus.

3.5. Experiment: Japanese Sentence Simplification

In order to confirm the language-independence of the proposed method, we also experiment in Japanese.

Table 3.6: Examples of English text simplification.

Input	Offenbach's numerous operettas, such as *Orpheus in the Underworld*, and *La belle Hélène*, were extremely popular in both France and the English-speaking world during the 1850s and 1860s.
Reference 1	Offenbach's ~~numerous~~ operettas, such as *Orpheus in the Underworld*, and *La belle Hélène*, were ~~extremely~~ **very** popular in ~~both~~ France and the English-speaking world during the 1850**'**s and 1860**'**s.
Reference 2	Offenbach's ~~numerous~~ **many** operettas, such as *Orpheus in the Underworld*, and *La belle Hélène*, were ~~extremely~~ **very** popular in ~~both~~ France and **in** the English-speaking world during the 1850s and 1860s.
Zhu et al. corpus	Offenbach's ~~numerous~~ **many** operettas, such as *Orpheus in the Underworld*, and *La belle Hélène*, were extremely popular ~~in~~ both France and the English-speaking ~~world during~~ **in** the 1850s and 1860s.
Coster and Kauchak corpus	Offenbach's ~~numerous~~ **many** operettas, such as *Orpheus in the Underworld*, and *La belle Hélène*, were extremely popular ~~in~~ both **in** France and the English-speaking ~~world during~~ **in** the 1850s and 1860s.
Hwang et al. corpus	Offenbach's ~~numerous~~ **many** operettas, such as *Orpheus in the Underworld*, and *La belle Hélène*, were extremely popular in both France and the English-speaking world during the 1850s and 1860s.
Our parallel corpus	Offenbach's ~~numerous~~ **many** operettas, such as *Orpheus in the Underworld*, and *La belle Hélène*, were ~~extremely~~ **very** popular in both France and the English-speaking world during the 1850s and 1860s.
Our pseudo-parallel corpus	Offenbach's ~~numerous~~ **many** operettas, such as *Orpheus in the Underworld*, and *La belle Hélène*, were extremely popular in ~~both~~ France and the English-speaking world during the 1850s and 1860s.

Table 3.7: Results of Japanese text simplification.

Threshold for MAS	# sents.	# rules	Avg. word difficulty level	BLEU	SARI
Baseline	0	0	2.56	58.5	24.3
MAS ≥ 0.80	867	14,356	2.53	52.9	30.9
MAS ≥ 0.75	2,299	49,688	2.54	46.5	32.1
MAS ≥ 0.70	11,941	307,275	2.51	40.1	34.4
MAS ≥ 0.65	90,542	2,609,793	2.49	28.6	31.7
MAS ≥ 0.60	470,885	15,128,855	2.51	19.9	28.3

3.5.1 Japanese Pseudo-Parallel Corpus for Text Simplification

We use Balanced Corpus of Contemporary Written Japanese [67] as a raw corpus. We use the average value of the word difficulty level[13] as a readability measure and CBOW model of the word2vec [73] on the BCCWJ corpus for sentence alignment. In order to reduce noise, only word pairs with a word similarity $[0.5, 1.0]$ are used for word alignment. As a result, 470,885 sentence pairs with a sentence similarity of $[0.6, 1.0)$ are extracted greedily into our pseudo-parallel corpus.

3.5.2 Settings

We experimented using PBSMT with the same settings as English. For evaluation, we used 2,227 sentence pairs. This sentence pairs were manually annotated to a comparable corpus crawled from the Web[14]. Two annotators gave simplification labels[15] (Good, Good Partial, Partial, Bad) to each sentence pair with MAS similarity $[0.75, 1.0)$. We used only 2,000 sentence pairs to which both annotators labeled Good or Good Partial.

3.5.3 Results

Table 3.7 shows the text simplification performance in Japanese. We only presented the baseline which does not perform any simplification because there is no parallel

[13] https://github.com/tmu-nlp/simple-jppdb

[14] https://matcha-jp.com/

[15] Pearson correlation coefficient reaches 0.769.

corpus for text simplification in Japanese. As with the experimental results in English, training with more corpus can acquire more simplification rules and generate simpler sentences. The reason why BLEU is lower than the experimental results in English is that it is being evaluated with the single reference test data. As we can see, our model improved SARI in compensation with BLEU as well as in English, we confirmed the language-independence of the proposed method.

In this chapter, we explained text simplification from only a raw corpus using PB-SMT. Previously, text simplification has primarily been studied in English for which rich language resources exist. In contrast, as a large-scale raw corpus can be used for many languages, this work opens the door to text simplification for many other languages.

Chapter 4

Further Improvement

We have done text simplification in the lexical substitution approach and the monolingual translation approach for Japanese that cannot use large-scale simplified corpora. In this chapter, we further improve the text simplification through experiments in English.

In Section 4.1, we improve paraphrase lexicon. This helps to acquire better candidates in the step of simplification candidates acquisition in the lexical substitution approach.

In Section 4.2, we improve the sentence similarity based on alignment between word embeddings according to the given corpus. This helps to perform better sentence alignment when building a parallel corpus in the monolingual translation approach.

In Section 4.3, we improve the automatic evaluation metrics for text simplification. In the past, output sentences were automatically evaluated by comparing output sentence and reference(s) in both lexical substitution approach and monolingual translation approach. Therefore, the performance of the text simplification system could only be evaluated on the specific datasets. In this work, we propose a novel referenceless automatic evaluation metrics for text simplification.

4.1. Improving Paraphrase Lexicon

Paraphrases are useful for flexible language understanding in many NLP applications. For example, the usefulness of PPDB [33, 86], a publicly available large-scale resource for lexical paraphrasing, has been reported for tasks like learning word embeddings [126] and semantic textual similarity [104]. In PPDB, paraphrase pairs are

acquired via word alignment on a bilingual corpus by a process called bilingual pivoting [10].

Although bilingual pivoting is widely used for paraphrase acquisition, it always includes noise due to unrelated word pairs caused by word alignment errors on the bilingual corpus. Distributional similarity, another well-known method for paraphrase acquisition, is free from alignment errors but also includes noise due to antonym pairs that share the same contexts on the monolingual corpus [77].

In this study, we formalize paraphrasability of paraphrase pairs acquired via bilingual pivoting using pointwise mutual information (PMI) and reduce the noise by reranking the pairs using distributional similarity. The proposed method extends Local PMI [31], which is a variant of weighted PMI that aims to avoid low-frequency bias in PMI, for paraphrase acquisition. Since bilingual pivoting and distributional similarity have different advantages and disadvantages, we combine them to construct a complementary paraphrase acquisition method, called MIPA.

Levy and Goldberg [65] explained a well-known representation learning method for word embeddings, the skip-gram with negative-sampling (SGNS) [73], as a matrix factorization of a word-context co-occurrence matrix with shifted positive PMI. In this study, we explained a well-known method for paraphrase acquisition, bilingual pivoting [10, 33], as a (weighted) PMI.

4.1.1 Bilingual Pivoting and MIPA

As described in Sections 2.2.3 and 2.2.4, we propose a MIPA score which improves bilingual pivoting [10]. The paraphrase probability defined by bilingual pivoting is as follows.

$$\begin{aligned} p(e_2 \mid e_1) &= \textstyle\sum_f p(e_2 \mid f, e_1)\, p(f \mid e_1) \\ &\approx \textstyle\sum_f p(e_2 \mid f)\, p(f \mid e_1) \end{aligned} \tag{4.1}$$

First, in order to deal with word alignment errors, we reduce the influence of high frequency words.

$$\begin{aligned} \mathrm{BPMI}(e_1, e_2) &= \log p(e_2 \mid e_1) + \log p(e_1 \mid e_2) - \log p(e_1) - \log p(e_2) \\ &= \log \frac{p(e_2 \mid e_1)}{p(e_2)} + \log \frac{p(e_1 \mid e_2)}{p(e_1)} = 2\mathrm{PMI}(e_1, e_2) \end{aligned} \tag{4.2}$$

Next, in order to deal with low frequency bias of pointwise mutual information, we weight it.

$$\mathrm{MIPA}(e_1, e_2) = \cos(\vec{e}_1, \vec{e}_2) \cdot \mathrm{BPMI}(e_1, e_2) \tag{4.3}$$

4.1.2 Settings

We used French-English parallel data[1] from Europarl-v7 [55] and GIZA++ [79] to calculate the conditional paraphrase probability $p(e_2 \mid e_1)$ and $p(e_1 \mid e_2)$. We also used the English Gigaword 5th Edition[2] and KenLM [42] to calculate the word probability $p(e_1)$ and $p(e_2)$. For $\cos(\vec{e}_1, \vec{e}_2)$, we used the cbow model[3] of word2vec [73]. Finally, we acquired paraphrase candidates of 170,682,871 word pairs except for paraphrase of itself ($e_1 = e_2$).

We employed the conditional paraphrase probability of bilingual pivoting given in Equation (4.1), symmetric paraphrase score of PPDB given by Equation (2.5) and distributional similarity as baselines, and compared them with PMI shown in Equation (4.2) and MIPA score given in Equation (4.3).

4.1.3 Evaluation Datasets and Metrics

For evaluation, we use two datasets included in Human Paraphrase Judgments[4] published by Pavlick et al. [86].

First, Human Paraphrase Judgments include a paraphrase list of 100 words or phrases randomly extracted from Wikipedia and processed using a five-step manual evaluation for each paraphrase pair (HPJ-Wikipedia). The correct paraphrase is a word that gained 3 or more evaluations in manual evaluation. We use this dataset to evaluate the acquired paraphrase pairs by Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP) following Pavlick et al. [86]. Furthermore, we evaluate the coverage of the top-k paraphrase pairs. Function words such as "as" have more than 50,000 types of paraphrase candidates because they are sensitive to word alignment errors in bilingual pivoting. However, since many of these paraphrase candidates are word pairs that are

[1] http://www.statmt.org/europarl/

[2] https://catalog.ldc.upenn.edu/LDC2011T07

[3] https://code.google.com/archive/p/word2vec/

[4] http://www.seas.upenn.edu/~epavlick/data.html

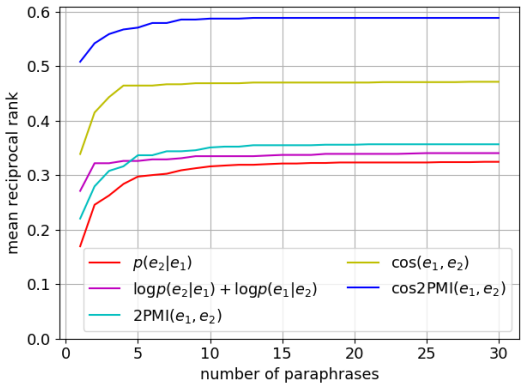

Figure 4.1: Paraphrase ranking in MRR.

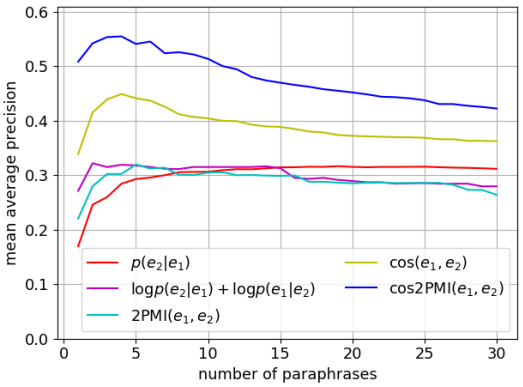

Figure 4.2: Paraphrase ranking in MAP.

not actually paraphrases, we evaluate the coverage of how they can reduce unnecessary candidates while preserving the correct paraphrases.

Second, Human Paraphrase Judgments also includes a five-step manual evaluation of 26,456 word pairs sampled from PPDB [33] (HPJ-PPDB) along with the paraphrase list of 100 words. We use this dataset to evaluate the overall paraphrase ranking based on Spearman's correlation coefficient as in Pavlick et al. [86].

4.1.4 Results

Figures 4.1 and 4.2 show the comparison of paraphrase rankings in MRR and MAP on HPJ-Wikipedia. The horizontal axis of each graph indicates the evaluation of the paraphrase up to the top-k of the paraphrase score. The evaluation in MRR in Figure 4.1 shows that the ranking performance of paraphrase pairs improves by making bilingual pivoting symmetric. PMI slightly outperforms the baselines of bilingual pivoting below the top 5. Furthermore, MIPA shows the highest performance because reranking by distributional similarity greatly improves bilingual pivoting.

Evaluation using MAP, shown in Figure 4.2, also reinforces the same result, i.e., reranking by distribution similarity improved bilingual pivoting, and MIPA showed the highest performance.

Figure 4.3 shows the coverage of the top-ranked paraphrases on HPJ-Wikipedia. Despite the fact that the symmetric paraphrase score outperforms conditional paraphrase probability in the ranking performance of the top three in MRR and MAP, it shows poor performance in terms of coverage. As also shown in Table 4.2, the symmetric

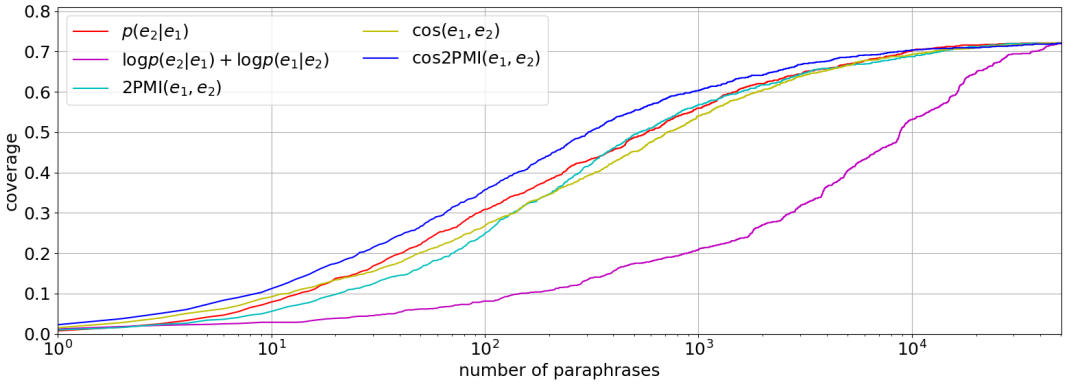

Figure 4.3: Coverage of the top-k paraphrase pairs.

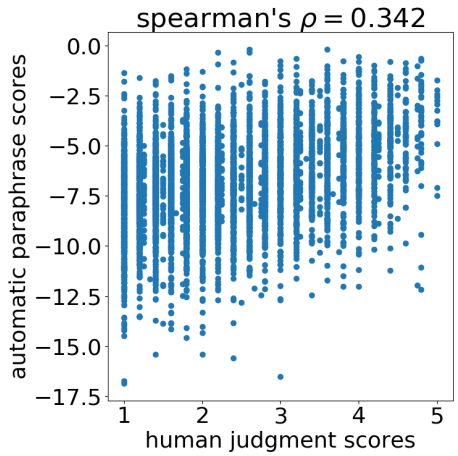

Figure 4.4: $\rho : \log p(e_2 \mid e_1)$

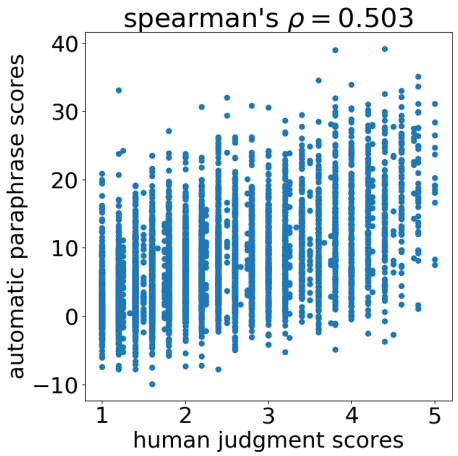

Figure 4.5: $\rho : \text{MIPA}(e_1, e_2)$

paraphrase score ranked the paraphrase with extremely high likelihood to 1st place, while the other rankings are unreliable. Although there is not much difference in other methods, MIPA outperforms the other methods.

Figures 4.4 and 4.5 show the scatter plots and Spearman's correlation coefficient of each paraphrase score and manual evaluation (average value of five evaluators) on HPJ-PPDB. As with the previous experimental results, MIPA showed the high correlation. Especially, the noise generated by false positives at the upper left of the scatter plot can be reduced by combining PMI and distributional similarity.

4.1.5 Extrinsic Evaluation

Next, we applied the acquired paraphrase pairs to the semantic textual similarity (STS) task and evaluated to what extent acquired paraphrases improve downstream applications. The semantic textual similarity task deals with calculating the semantic similarity between two sentences. In this study, we evaluate by Pearson's correlation coefficient with five-step manual evaluation using five datasets constructed by SemEval [5, 6, 3, 2, 4]. We applied the acquired paraphrase pairs to the unsupervised method of DLC@CU [104], which achieved excellent results using PPDB in the semantic textual similarity task of SemEval-2015 [2]. DLS@CU performs word alignment [103] using PPDB, and calculates sentence similarity according to the ratio of aligned words as follows.

$$\begin{aligned} \mathrm{sim}(x,y) &= \frac{\mathrm{PA}(x,y)+\mathrm{PA}(y,x)}{|x|+|y|} \\ \mathrm{PA}(x,y) &= \sum_{i=1}^{|x|} \begin{cases} 1 & \exists j : x_i \Leftrightarrow y_j \in y \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4.4}$$

where $x_i \Leftrightarrow y_j$ holds if and only if the word pair (x_i, y_j) is included in a given paraphrase lexicon. Although DLS@CU targets all paraphrases of PPDB, we only use the top 10 words of paraphrase score for each target word and compare the performance of paraphrase scores.

Table 4.1 shows the experimental results of the semantic textual similarity task. "ALL" is the weighted mean value of the Pearson's correlation coefficient over the five datasets. MIPA achieved the highest performance with three out of five datasets. In other words, the proposed method extracted paraphrase pairs useful for calculating sentence similarity at the top-rank.

Table 4.1: Evaluation by Pearson's correlation coefficient in STS task.

	$p(e_2 \mid e_1)$	BP(e_1, e_2)	BPMI(e_1, e_2)	$\cos(\vec{e}_1, \vec{e}_2)$	MIPA(e_1, e_2)
STS-2012	**0.539**	0.466	0.383	0.363	0.442
STS-2013	0.489	0.469	0.463	0.483	**0.499**
STS-2014	0.464	0.460	0.471	0.453	**0.475**
STS-2015	0.611	0.655	0.660	0.642	**0.671**
STS-2016	0.444	0.518	**0.550**	0.518	0.542
ALL	0.536	0.543	0.534	0.523	**0.555**

4.1.6 Examples

Table 4.2 shows examples of the top 10 in paraphrase rankings. In the paraphrase examples of *cultural*, conditional paraphrase probability does not score the correct paraphrase as top-ranked words. Although symmetric paraphrase score ranked the correct paraphrase at the top, words other than the top are less reliable as shown by the previous experimental results. PMI is strongly influenced by low-frequency words, and many of the top-ranked words are singleton words in the bilingual corpus. MIPA, in contrast, mitigates the problem of low-frequency bias, and many of the top-ranked words are correct paraphrases. Distributional similarity-based methods include relatively many of correct paraphrases at the top, and the other top-ranked words are also strongly related to *cultural*. From the viewpoint of paraphrase, three words out of the top 10 words of the proposed method are incorrect, but these words may also be useful for applications such as learning word embeddings [126] and semantic textual similarity [104].

Table 4.3 shows correct examples of the paraphrase rankings. In the paraphrase examples of *labourers*, there were 20 correct paraphrases that received a rating of 3 or higher in manual evaluation. With respect to the conditional paraphrase probability and PMI, it is necessary to consider up to the 400th place to cover all correct paraphrases. On the other hand, distributional similarity-based methods have correct paraphrases of relatively higher rank. In particular, MIPA was able to include 10 words of correct paraphrases in the top 20 words. In other words, the proposed method can obtain paraphrase with high coverage by using only the highly ranked paraphrases.

Table 4.2: Paraphrase examples of *cultural*. Italicized words are the correct words.

	$p(e_2 \mid e_1)$	BP(e_1, e_2)	BPMI(e_1, e_2)	cos$(\vec{e}_1, \vec{e}_2)$	MIPA(e_1, e_2)
1.	diverse	*culturally*	culturally-based	historical	*socio-cultural*
2.	harvests	*culture*	culturaldevelopment	*culture*	*culture*
3.	firstly	151	cultural-social	educational	*multicultural*
4.	understand	charter	economic-cultural	linguistic	*intercultural*
5.	flowering	monuments	culture-	*multicultural*	educational
6.	trying	art	cultural-educational	*cross-cultural*	intellectual
7.	structure	casal	kulturkampf	diversity	*culturally*
8.	january	kahn	cultural-political	technological	*sociocultural*
9.	*culture*	13	multiculture	intellectual	*heritage*
10.	*culturally*	caning	*culturally*	preservation	architectural

Table 4.3: Correct paraphrase examples of *labourers*.

$p(e_2 \mid e_1)$		BP(e_1, e_2)		BPMI(e_1, e_2)		cos$(\vec{e}_1, \vec{e}_2)$		MIPA(e_1, e_2)	
1.	workers	9.	gardeners	10.	workmen	2.	workers	2.	workers
2.	employees	42.	harvesters	11.	wage-earners	8.	people	4.	workmen
9.	farmers	62.	workers	16.	earners	10.	persons	5.	craftsmen
13.	labour	71.	seafarers	19.	workers	11.	farmers	6.	wage-earners
16.	gardeners	73.	unions	21.	craftsmen	15.	craftsmen	9.	persons
17.	people	99.	homeworkers	22.	workforces	26.	wage-earners	12.	employees
28.	workmen	283.	works	26.	employed	27.	workmen	13.	earners
30.	employed	394.	workmen	27.	employees	29.	harvesters	15.	farmers
33.	craftsmen	395.	employees	50.	labour	31.	seafarers	18.	people
59.	harvesters	412.	wage-earners	55.	persons	32.	employees	19.	workforces
80.	work	415.	craftsmen	75.	farmers	42.	gardeners	37.	harvesters
88.	earners	417.	earners	103.	homeworkers	47.	earners	42.	individuals
90.	wage-earners	419.	labour	105.	individuals	55.	workforces	53.	labour
106.	persons	420.	employed	112.	work	57.	individuals	55.	seafarers
109.	individuals	431.	people	135.	people	79.	unions	65.	gardeners
114.	seafarers	433.	farmers	187.	harvesters	103.	labour	88.	employed
115.	unions	446.	workforces	273.	gardeners	140.	homeworkers	100.	homeworkers
131.	workforces	451.	work	317.	seafarers	144.	work	105.	work
166.	homeworkers	453.	persons	456.	unions	170.	employed	149.	unions
401.	works	474.	individuals	469.	works	222.	works	254.	works

4.2. Improving Sentence Similarity Measurement

Measuring the similarity between short textual units such as sentences, tweets or chat messages is a commonplace task in numerous natural language processing (NLP) applications such as information retrieval, text clustering, or classification. Compared to measuring the similarity between longer textual units such as documents that contain many words, measuring the similarity between short sentences is a challenging task due to the lack of common features. Consequently, similarity measures based on word overlap such as cosine similarity, often fails to detect the similarity between sentences [5]. To overcome this feature sparseness problem, prior work on sentence similarity have proposed methods that use external lexical resources such as thesauri [109], or project sentences into a lower-dimensional dense spaces in which subsequently similarity is computed [36, 50, 45, 125, 62, 53].

We propose a complementary approach for measuring the similarity between two sentences in a corpus that considers not only the features that occur in those two sentences, but also features that occur in *all* pairs of sentences in the corpus. Specifically, we require sentence similarity scores to satisfy two important types of constraints: (A) if two sentences share many common features, then it is likely that the remaining features in each sentence are also related, and (B) if two sentences contain many related features, then those two sentences are themselves similar.

To motivate the role played by these constraints consider the following three example sentences.

(i) *I love dogs and cats.*

(ii) *I love dogs and rabbits.*

(iii) *My favorite pet is a cat.*

Sentences (i) and (ii) share many common content words such as *I*, *love*, and *dog*. Thus, we can infer that *cat* and *rabbit* must also be semantically related. The confidence of our inference grows with (a) the proportion of the overlap, and (b) the number of different sentence pairs in which we observe similar overlaps. Consider now that we are further required to compare sentences (ii) and (iii), between which we have no common words. Without the knowledge that *cat* and *rabbit* are related from our previous comparison, we would predict a zero similarity score between sentences (ii) and (iii). However, if we use the knowledge obtained from (i) and (ii), and consider *cat* and *rabbit* to be similar (i.e. pets in this case), then we could predict a non-zero

similarity score for (ii) and (iii). Therefore, we can benefit from the constraints derived from other pairs of sentences in a corpus (such as (i) and (ii)), when measuring the similarity between two given sentences selected from that corpus (such as (ii) and (iii)).

Our proposed method iterates over two stages.

- First, we align each sentence in a corpus with all the other similar sentences to build a word-alignment matrix. We compute the similarity between two words based on two factors: (a) pointwise mutual information between the two words according to their alignment frequency in the word-alignment matrix, and (b) prior similarity between words measured using pre-trained word embeddings. Using the computed word similarity scores, we measure the similarity between two sentences using three sentence alignment methods.

- Second, we update the word similarity scores using the word-alignment matrix computed in the first stage. Specifically, we propose two update rules for this purpose: an additive update, and a multiplicative update. The proposed method iterates multiple times over the corpus measuring similarities between all pairs of sentences. In practice, the proposed method converges in less than 3 iterations. However, computing all sentence pair similarities can be time consuming for large text corpora. To overcome this problem, we propose an efficient method to identify the top-most similar sentence pairs in a corpus that contribute to the similarity score update using SimHash [89] that obviates all-pair comparisons.

Our proposed method is unsupervised in the sense that it does not require any labeled data for sentence similarity. Moreover, we do not use external resources such as thesauri, which might not be available for resource poor languages or specialised domains. The proposed method does not assume a specific sentence representation method, and can be used with different sentence representations such as bag-of-words, or parse trees. Moreover, it is complementary to the sentence embedding methods, and can be used in conjunction in an ensemble setting as yet another sentence similarity measure.

We evaluate the proposed sentence similarity method using the SemEval-2015 Task 2 sentence similarity benchmark dataset. Our experimental results show that the proposed iterative approach for measuring sentence semantic similarity is significantly better than the non-iterative counterparts.

4.2.1 Iterative Similarity Computation

Our proposed method iterates between two stages. First, we use the similarity between words to align pairs of sentences in a corpus. Following Song and Roth [97], we extend three sentence similarity measures for iterative similarity computation. Second, we update the word similarity scores considering the sentence alignments produced in the first stage. Two update rules are proposed for this purpose.

Sentence Alignment

As in Section 3.2, we perform sentence alignment using three types of sentence similarity AAS, MAS, and HAS. AAS and MAS average the word similarities $\phi(x_i, y_j)$ between pairs of words within given two sentences, x and y. AAS and MAS deal with many-to-many and one-to-many word alignments, respectively. On the other hand, HAS is based on one-to-one word alignments. The task of identifying the best one-to-one word alignments $\mathscr{H}$ is regarded as a problem of bipartite graph matching, where the two sets of vertices respectively comprise words within each sentence x and y, and the weight of a edge between x_i and y_j is given by the word similarity. Given $\mathscr{H}$ identified using the Hungarian algorithm [58], HAS is computed by averaging the word similarities $\phi(x_i, y_j)$ between the aligned pairs of words.

$$\mathrm{AAS}(x,y) = \frac{1}{|x||y|} \sum_{i=1}^{|x|} \sum_{j=1}^{|y|} \phi(x_i, y_j) \tag{4.5}$$

$$\mathrm{MAS}(x,y) = \frac{1}{2|x|} \sum_{i=1}^{|x|} \max_j \phi(x_i, y_j) + \frac{1}{2|y|} \sum_{j=1}^{|y|} \max_i \phi(x_i, y_j) \tag{4.6}$$

$$\mathrm{HAS}(x,y) = \frac{1}{|\mathscr{H}|} \sum_{(i,j)\in\mathscr{H}} \phi(x_i, y_j) \tag{4.7}$$

where $|\mathscr{H}| = \min(|x|, |y|)$, as $\mathscr{H}$ contains only one-to-one word alignments.

Incremental Update Rule

In many text similarity computation tasks such as finding similar documents in information retrieval, or document clustering, we must compare not only one pair of texts (documents) selected from a given collection, but compute the similarities between all

pairs of texts. Likewise, when calculating the similarity between sentences, it is often the case that we are given a large collection of sentences (a corpus) from which a pair of sentences is selected. As we already described, we can exploit the information available in all the sentences in the corpus when measuring the similarity between two given sentences. Instead of considering the similarity between two words, $\phi(x_i, y_j)$, to be a fixed value, we update word similarities considering their alignments in sentences. Because the sentence similarity measures given by (4.5), (4.6), and (4.7) depend on the word similarity scores, this results in an update procedure that iterates between measuring sentence similarities (thereby word-alignments), and updating word similarity scores.

Let us denote the similarity between two words x_i and y_j after the t-th iteration by $\phi^{(t)}(x_i, y_j)$, and the word-alignment matrix computed using MAS or HAS by $\mathcal{A}^{(t)}$. Note that the word-alignment matrix $\mathcal{A}$ is an asymmetric matrix. Therefore, we define a symmetric word co-occurrence matrix $\mathcal{C}^{(t)}$, where its (i, j)-th element is given by:

$$\mathcal{C}_{ij}^{(t)} = (\mathcal{A}_{ij}^{(t)} + \mathcal{A}_{ji}^{(t)})/2 \tag{4.8}$$

Let $\mathcal{B}^{(t)}$ be the word similarity matrix where its (i, j) element $\mathcal{B}_{ij}^{(t)}$ denotes the similarity between the two words i and j computed using co-occurrence counts $\mathcal{C}_{ij}^{(t)}$. Different word association measures can be used to compute similarity scores from co-occurrence counts. In this work, we use the positive pointwise mutual information (PPMI) [78] computed as follows:

$$\mathcal{B}_{ij}^{(t)} = \max\left(0, \log\left(\frac{\mathcal{C}_{ij}^{(t)} \times \sum_{ij} \mathcal{C}_{ij}^{(t)}}{\sum_i \mathcal{C}_{ij}^{(t)} \sum_j \mathcal{C}_{ij}^{(t)}}\right)\right) \tag{4.9}$$

PPMI is frequently used for measuring word similarity in various NLP tasks [110].

We propose two update rules for updating the word similarity scores using the word-alignment counts: the *additive update rule* defined by (4.10), and the *multiplicative update rule* defined by (4.11).

$$\phi^{(t+1)}(x_i, y_j) = \phi^{(t)}(x_i, y_j) + \eta^{(t)} \mathcal{B}_{ij}^{(t)} \tag{4.10}$$

$$\phi^{(t+1)}(x_i, y_j) = \phi^{(t)}(x_i, y_j) \mathcal{B}_{ij}^{(t)} \tag{4.11}$$

Here, $\eta^{(t)}$ is the update rate in the t-th iteration. Because we require word similarity scores to be in the range $[0, 1]$, we scale $\phi^{(t+1)}(x_i, y_j)$ by dividing from the maximum

similarity score between any pair of words, $\max_{ij} \phi^{(t+1)}(x_i, y_j)$, after each iteration. In both update rules, the initial word similarities, $\phi^{(0)}(x_i, y_j)$, are computed using pre-trained word embeddings. In our experiments, we used skip-gram with negative sampling (SGNS) [73] for learning word embeddings. Then, $\phi^{(0)}(x_i, y_j)$ is computed as the cosine similarity between the word embeddings corresponding to the words x_i and y_j.

The additive update rule given by (4.10) closely resembles the update rule used in imitation learning [91], where a learner is required to imitate the training signal provided by an oracle. In our case, the word similarity scores $\phi^{(t)}(x_i, y_j)$ are required to follow $\mathscr{B}_{ij}^{(t)}$, the similarity scores computed using word-alignment counts. On the other hand, the multiplicative update rule given by (4.11) can be seen as a weighted similarity score where current similarity scores are weighted by the corresponding alignment counts. We experimentally compare the different combinations of word-alignment matrices produced by different sentence similarity measures and the update rules.

In practice, even though two sentences might be similar, not all the words in the two sentences need to be similar. However, both MAS and HAS require all word-pairs from the two sentences to be aligned. This imposes an unnecessarily strict constraint on word-alignment because two words might get aligned despite having a small word similarity score. To avoid such word-alignments, we consider only word-pairs (x_i, y_j) with similarity $\phi^{(t)}(x_i, y_j) > \theta$ for the word-alignment process for a fixed threshold $\theta \in [0, 1]$. We experimentally study the effect of θ on the performance of our method.

Efficient Computation of Similarity

Calculating the full word-alignment matrix requires computational complexity of $\mathscr{O}(n^2|\mathscr{V}|)$, where n is the total number of sentences in the corpus. However, most sentence pairs in a corpus will have almost zero similarity scores, and would not contribute to the word-alignment matrices. To avoid such unproductive computations, we use SimHash [89] to find the most similar k sentences for each sentence in the corpus, and measure sentence similarity only for those sentence pairs. Hamming distance over SimHash values of two sentences approximates the cosine similarity between the corresponding sentences. This method reduces the computational complexity to $\mathscr{O}(nk|\mathscr{V}|)$, which is significantly smaller than $\mathscr{O}(n^2|\mathscr{V}|)$ for $k \ll n$.

4.2.2 Settings

For evaluating the proposed method for measuring sentence similarity, we use the SemEval-2015 Task 2 dataset[5] [2]. This dataset includes 3,000 sentence pairs from five different domains: news headlines (Head), image descriptions (Img), answer pairs from a tutorial dialogue system (Stud), answer pairs from Q&A websites (QA), and sentence pairs from a committed belief dataset (Bel). Sentence similarity scores that range between 0 (the two sentences are completely dissimilar) to 5 (the two sentences are completely equivalent, as they mean the same thing) are obtained via crowdsourcing. A sentence similarity measure is evaluated against the human ratings in this dataset using the Pearson correlation coefficient. Pearson correlation coefficient ranges in $[-1, 1]$, and high values indicate better agreement with the human notion of sentence similarity.

We use publicly available pre-trained word embeddings[6] trained using SGNS and use cosine similarity to compute initial word similarities, $\phi^{(0)}(x_i, y_j)$, required by the additive and the multiplicative rules defined respectively by (4.10) and (4.11). The pre-trained word embeddings are trained on about 100 billion word Google News corpus, and 300 dimensional vectors for 3 million words are created. We use 5-fold cross validation on the train sentence pairs in the SemEval-2015 Task 2 dataset to obtain the optimal values of $\theta = 0.4$ and $t = 3$. Moreover, we experimented with different learning rate scheduling methods and found $\eta^{(t)} = 1$ to be the best. We analyse the sensitivity of the performance of the proposed method to those parameters. Because the SemEval-2015 Task 2 dataset contains only a small number of sentences (ca. $6,000$), we do not require the SimHash-based approximation method for this dataset.

To demonstrate the effectiveness of conducting iterative similarity updates in the proposed method, we compare it against the following baseline methods that have been frequently used in prior work that do not perform iterative similarity updates.

Cosine baseline calculates the similarity between two sentences x and y as the cosine similarity between the two vectors $\vec{x}$ and $\vec{y}$ representing the two sentences.

Cosine (add SGNSs) baseline calculates the similarity between two sentences x and y as the cosine similarity between two sentence embeddings. These sentence embeddings are composed by adding the word embeddings of the words in each

[5] http://alt.qcri.org/semeval2015/task2/

[6] https://code.google.com/archive/p/word2vec/

sentence. Representing sentences via the sum of word embeddings has been shown to be a strong baseline for creating sentence embeddings [43].

SGNS method calculates the similarity between two sentences x and y using the three sentence similarity measures, AAS, MAS, and HAS respectively using (4.5), (4.6), and (4.7). It uses the pre-trained word embeddings learnt using SGNS, and measures the similarity $\phi(x_i, y_j)$, between two words x_i and y_j as the cosine similarity between the corresponding word embeddings. This method simulates the proposals made by Song and Roth [97] for measuring sentence similarity using word alignments. This method *does not* perform any iterative similarity updates as done by the proposed method, and corresponds to the current state-of-the-art unsupervised sentence similarity measure.

PPMI baseline uses the PPMI-based word similarity computed using word-alignment counts, as the word similarity function $\phi(x_i, y_j)$, and computes the three sentence similarity measures AAS, MAS, and HAS. Specifically, 6 variants of this baseline is computed by combining the two word-alignment matrices $\mathscr{A}_{\text{MAS}}$, and $\mathscr{A}_{\text{HAS}}$, with the three sentence similarity measures AAS, MAS, and HAS.

4.2.3 Results

Table 4.4 compares the different sentence similarity measures using the Pearson correlation coefficients with the human ratings for the test sentence pairs in the SemEval-2015 Task 2 dataset. The proposed method (denoted by **Prop**) is computed for the combinations of 2 word-alignment matrices ($\mathscr{A}_{\text{MAS}}$ and $\mathscr{A}_{\text{HAS}}$), 3 sentence similarity measures (AAS, MAS, and HAS), and 2 update rules (additive and multiplicative, denoted respectively by $+$ and $*$), resulting in 12 variants shown in Table 4.4. The final column, **Mean**, in Table 4.4 shows the weighted mean over the 5 domains for each method. It is computed by weighting the Pearson correlation coefficient in each domain by the total number of sentence pairs in that domain, according to the official scoring guidelines in SemEval-2015 Task 2.

As mentioned in Section 3.2.2, AAS inevitably involves noise, as many word pairs are semantically irrelevant to each other. Therefore, the performance based on AAS is generally lower than that of MAS or HAS.

From Table 4.4, we see that **Prop** $\mathscr{A}_{\text{MAS}} + \text{MAS}$ is the best performing method among the different methods compared. In particular, it reports the best correla-

Table 4.4: Sentence similarity measurement results on the SemEval-2015 Task 2 dataset. The bold scores means the highest performance. The scores with a star statistically significantly outperform the SGNS (MAS) baseline.

	Head	Img	Stud	QA	Bel	Mean
Cosine	0.531	0.603	0.664	0.445	0.651	0.587
Cosine (add SGNSs)	0.567	0.531	0.620	0.296	0.465	0.525
SGNS AAS	0.294	0.316	0.043	0.079	0.125	0.189
SGNS MAS	0.603	0.626	0.656	0.391	0.636	0.599
SGNS HAS	0.590	0.614	0.682	0.386	0.615	0.596
PPMI $\mathscr{A}_{\text{MAS}}$ AAS	0.206	0.325	0.187	0.236	0.137	0.226
PPMI $\mathscr{A}_{\text{MAS}}$ MAS	0.540	0.561	**0.701**	0.327	0.591	0.565
PPMI $\mathscr{A}_{\text{MAS}}$ HAS	0.531	0.553	0.697	0.320	0.574	0.557
PPMI $\mathscr{A}_{\text{HAS}}$ AAS	0.340	0.368	0.327	0.370	0.221	0.333
PPMI $\mathscr{A}_{\text{HAS}}$ MAS	0.543	0.602	0.679	0.437	0.654	0.592
PPMI $\mathscr{A}_{\text{HAS}}$ HAS	0.533	0.586	0.675	0.430	0.634	0.582
Prop $\mathscr{A}_{\text{MAS}} + \text{AAS}$	0.456	0.401	0.374	0.477	0.255	0.399
Prop $\mathscr{A}_{\text{MAS}} + \text{MAS}$	**0.639**	**0.643**	0.674	**0.501**	**0.671**	**0.636***
Prop $\mathscr{A}_{\text{MAS}} + \text{HAS}$	0.626	0.629	0.674	0.491	0.654	0.626
Prop $\mathscr{A}_{\text{HAS}} + \text{AAS}$	0.443	0.398	0.361	0.450	0.254	0.388
Prop $\mathscr{A}_{\text{HAS}} + \text{MAS}$	0.638	0.642	0.673	0.498	0.670	0.634*
Prop $\mathscr{A}_{\text{HAS}} + \text{HAS}$	0.626	0.629	0.674	0.491	0.654	0.625
Prop $\mathscr{A}_{\text{MAS}} * \text{AAS}$	0.424	0.395	0.371	0.444	0.262	0.386
Prop $\mathscr{A}_{\text{MAS}} * \text{MAS}$	0.601	0.631	0.674	0.480	0.666	0.620
Prop $\mathscr{A}_{\text{MAS}} * \text{HAS}$	0.591	0.619	0.674	0.474	0.650	0.612
Prop $\mathscr{A}_{\text{HAS}} * \text{AAS}$	0.423	0.395	0.370	0.439	0.262	0.385
Prop $\mathscr{A}_{\text{HAS}} * \text{MAS}$	0.601	0.631	0.674	0.479	0.665	0.619
Prop $\mathscr{A}_{\text{HAS}} * \text{HAS}$	0.591	0.619	0.674	0.474	0.651	0.612

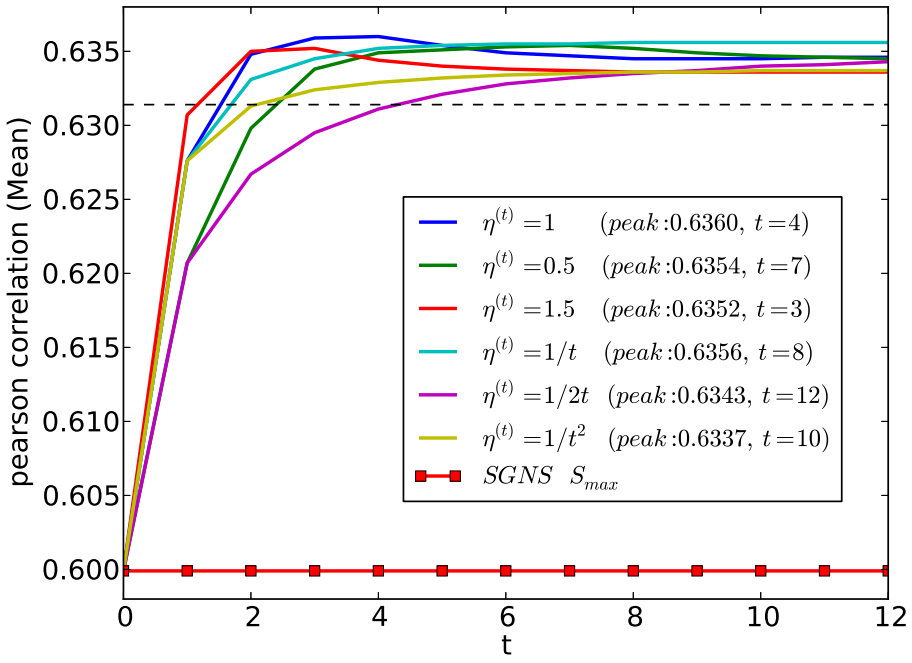

Figure 4.6: Effect of the different update rate scheduling methods on the performance of the proposed method is shown. The dashed horizontal line shows $p < 0.05$ significance level (Fisher z-transformation) for outperforming the **SGNS** MAS method. Peak correlation value and the required number of iterations (t) are shown within brackets.

tion coefficients in 4 out of the 5 domains. Moreover, according to the Fisher z-transformation, the correlations reported by the proposed method is statistically significantly better than that of **SGNS** MAS, which supports our proposal that sentence similarities must be computed in an iterative fashion over the entire corpus considering word-alignment constraints. Overall, the maximum word-alignment ($\mathscr{A}_{\text{MAS}}$) with MAS consistently perform well across different domains and baselines.

Between the two update rules, additive update outperforms the multiplicative counterpart. Recall that the word similarity matrix $\mathscr{B}^{(t)}$ given by (4.9) is in practice a sparse matrix. Therefore, the multiplicative update rule given by (4.11) results in even sparser similarity scores $\phi^{(t+1)}$ than $\phi^{(t)}$ after each update. On the other hand, the additive update rule given by (4.10) would retain the non-zero elements in $\phi^{(t)}$ during the update. We believe that the extra sparsification in the multiplicative update rule decreases its

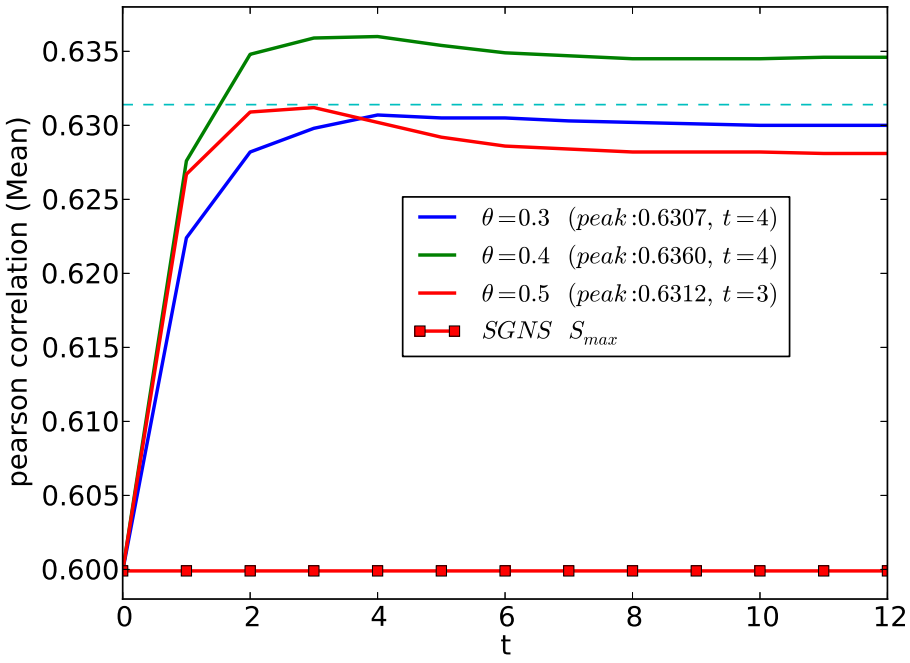

Figure 4.7: Effect of selecting word-pairs with similarity greater than θ for updating the word-alignment matrix. The dashed horizontal line shows $p < 0.05$ significance level (Fisher z-transformation) for outperforming the **SGNS** MAS method. Peak correlation value and the required number of iterations (t) are shown within brackets.

performance when measuring the sentence similarities.

4.2.4 Parameter Sensitivity

We study the performance of the **Prop** $\mathscr{A}_{\text{MAS}} + \text{MAS}$ method, which reported the best results according to Table 4.4, under different update rate scheduling methods. Specifically, we consider update rate scheduling methods frequently used in stochastic optimization such as constant update rates ($\eta^{(t)} = 0.5, 1.0, 1.5$), reciprocal update rates ($\eta^{(t)} = 1/t, 1/2t$), and the inverse squared update rate ($\eta^{(t)} = 1/t^2$).

Fig 4.6 shows the performance of the proposed method under different update rate scheduling methods. The dashed horizontal line in Fig 4.6 is the level of performance a particular method must obtain in order for that method to statistically significantly

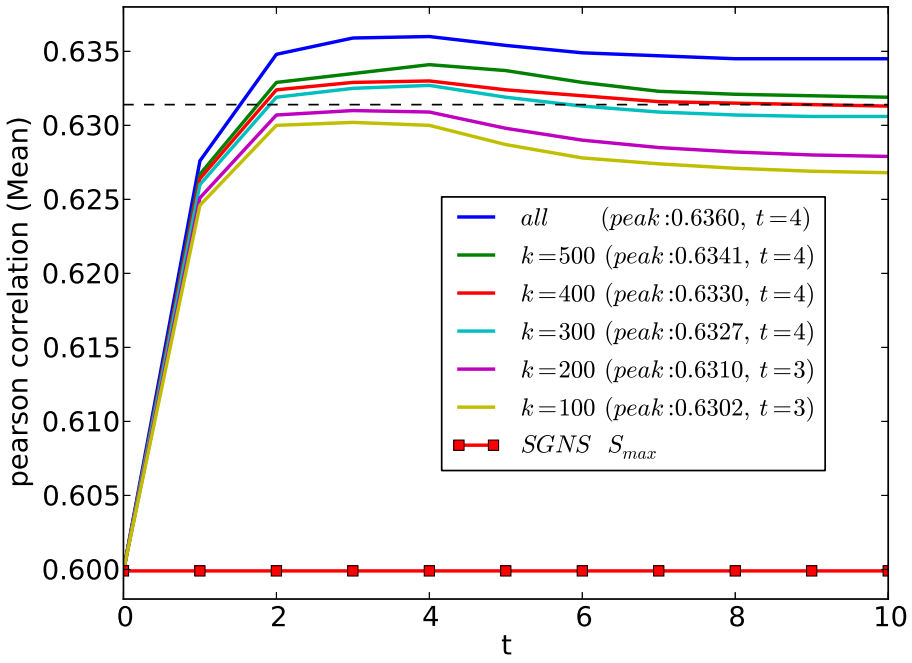

Figure 4.8: Effect of the number of top-k similar sentences selected using SimHash on the performance of the proposed method is shown. The dashed horizontal line shows $p < 0.05$ significance level (Fisher z-transformation) for outperforming the **SGNS** MAS method. Peak correlation value and the required number of iterations (t) are shown within brackets.

outperform the state-of-the-art **SGNS** MAS. From Fig 4.6, we see that our proposed method outperforms **SGNS** MAS under all update rate scheduling methods. Therefore, the proposed method is relatively insensitive to the update rate scheduling method used.

Moreover, under constant update rates, when we increase the value of η, the Pearson correlation reaches the maximum value with a smaller number of iterations. Once the Pearson correlation coefficients have reached these maximum values, the performance converges. Because it is desirable to converge to the best correlation value with smaller number of iterations, $\eta^{(t)} = 1.5$ (peak performance achieved after 3 iteration) is a suitable value.

Fig 4.7 shows the effect of considering word-pairs greater than similarity θ during the sentence similarity measurement process. Considering less similar word-pairs in

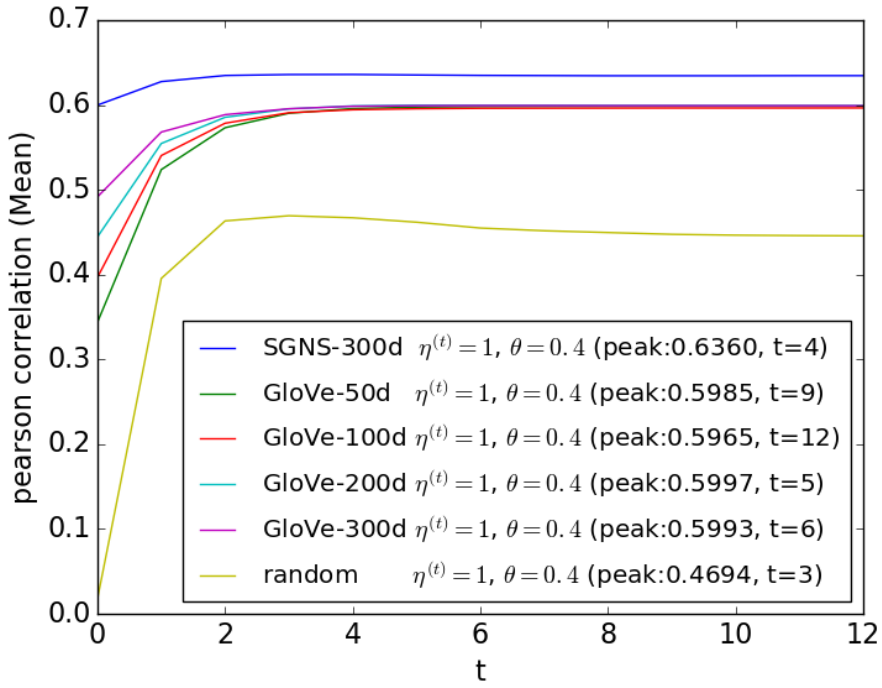

Figure 4.9: Effect of the different initial word embeddings on the performance of the proposed method is shown. The dashed horizontal line shows $p < 0.05$ significance level (Fisher z-transformation) for outperforming the **SGNS** MAS method. Peak correlation value and the required number of iterations (t) are shown within brackets.

the alignment step leads to poor performance because of noisy alignments. On the other hand, high θ values will limit the number of words that we align between two sentences, leading to feature sparseness issues. This trade-off can be seen from the three curves shown in Fig 4.7.

To study the effect of selecting top-k similar sentences using SimHash, in Fig 4.8 we measure the performance of **Prop** $\mathscr{A}_{\text{MAS}}$ + MAS against different k values. We see that even selecting a small sample as the top-most similar $k = 100$ sentences for each sentence in the corpus out of all sentences (ca. 6,000), the proposed method can obtain a high (0.6302) correlation coefficient. With $k = 300$ similar sentences we can obtain statistically significant improvements over **SGNS** MAS. This is attractive when computing sentence similarities in large corpora. For example, even for a small corpus such as the SemEval-2015 Task 2 dataset, which has only 6,000 sentences, time taken

Table 4.5: Sentence similarity results using Word Mover's Distance on the SemEval-2015 Task 2 dataset.

	Head	Img	Stud	QA	Bel	Mean
Euclidean	0.648	**0.607**	0.689	0.428	**0.552**	0.609
Prop (t=0)	0.635	0.588	**0.702**	0.477	0.520	0.606
Prop (t=1)	**0.651**	0.592	**0.702**	0.495	0.539	**0.615**
Prop (t=2)	**0.651**	0.592	0.698	**0.496**	0.544	**0.615**
Prop (t=3)	0.649	0.593	0.695	**0.496**	0.545	0.614

for one iteration is reduced from 24 min to 1.5 min, by using $k = 100$.

To demonstrate the effect of the different initial word embeddings, we initialize using random vectors, and publicly available pre-trained word embeddings: 300 dimensional SGNS vectors[6] for 3 million words, 50, 100, 200 and 300 dimensional GloVe vectors[7] for 400 thousand words. As shown in Fig 4.9, our proposed method can significantly improve any initial word similarity by iterative updating. The better performance of SGNS over GloVe can be explained by the larger vocabulary covered by SGNS.

4.2.5 Sentence Similarity Complement

We improve an existing sentence similarity measure by a combination with the proposed method. The Word Mover's Distance [60] which is a sentence similarity measure based on the dissimilarity between words is improved in this study.

Table 4.5 compares the different word dissimilarity measure for the Word Mover's Distance. **Euclidean** baseline is calculated by the Euclidean distance $\left|\left|\vec{x}_i - \vec{y}_j\right|\right|$ between word x_i and word y_j in the SGNS embeddings. **Prop** dissimilarity measure is calculated using our updated word similarity $1 - \phi^{(t)}(x_i, y_j)$. From Table 4.5, we can see that **Prop** method calculated using our updated word similarity improves Word Mover's Distance [60] calculated using **Euclidean** distance. We confirmed the improvement of performance even in a small dataset (QA) consisting only of 375 sentence pairs.

[7] http://nlp.stanford.edu/projects/glove/

Table 4.6: The QATS training data shows that typical MT metrics are strongly biased by the length difference between original and simple sentences (r_{length}), while they are less correlated with the manually-labeled quality (r_{label}).

Metrics	r_{length}	r_{label}
BLEU	-0.765	0.245
METEOR	-0.617	0.257
TER	0.741	-0.233
WER	0.757	-0.230

4.3. Improving Evaluation Metrics for Simplification

This section examines the usefulness of semantic features based on word alignments for estimating the quality of text simplification. Specifically, we introduce seven types of alignment-based features computed on the basis of word embeddings and paraphrase lexicons. Through an empirical experiment using the QATS dataset [117], we confirm that we can achieve the state-of-the-art performance only with these features.

Similarly to other text-to-text generation tasks, such as MT and summarization, the outputs of text simplification systems have been evaluated subjectively by humans [120, 114] or automatically by comparing with handcrafted reference texts [98, 27, 122]. However, the former is costly and not replicable, and the latter has achieved only a low correlation with human evaluation.

On the basis of this backdrop, Quality Estimation (QE) [100], i.e., automatic evaluation without reference, has been drawing much attention in the research community. In the shared task on quality assessment for text simplification (QATS),[8] two tasks have been addressed [117]. One is to estimate a real-value quality score for given sentence pair, while the other is to classify given sentence pair into one of the three classes (*good*, *ok*, and *bad*). In the classification task of the QATS workshop, systems based on deep neural networks [82] and MT metrics [116] have achieved the best performance. However, deep neural networks are rather unstable because of the difficulty of training on a limited amount of data; for instance, the QATS dataset offers only 505 sentence pairs for training. MT metrics are incapable of properly capturing deletions that are prevalent in text simplification [27], as they are originally designed to gauge semantic equivalence. In fact, as shown in Table 4.6, well-known MT metrics are strongly biased by the length difference between original and simple sentences, even though it is

[8] http://qats2016.github.io/shared.html

rather unrelated with the quality of text simplification assessed by humans.

In order to properly account for the surface-level inequivalency occurring in text simplification, we examine semantic similarity features based on word embeddings and paraphrase lexicons. Unlike end-to-end training with deep neural networks, we quantify word-level semantic correspondences using two pre-compiled external resources: (a) word embeddings learned from large-scale monolingual data and (b) a large-scale paraphrase lexicon. Using the QATS dataset, we empirically demonstrate that a supervised classifier trained upon such features achieves good performance in the classification task.

4.3.1 Semantic Features Based on Word Alignments

We bring a total of seven types of features that are proven useful in Chapter 3. Specifically, we assume that some of these features are useful to capture inequivalency between original sentence and its simplified version introduced during simplification, such as lexical paraphrases and deletion of words and phrases.

Throughout this subsection, original sentence and its simplified version are referred to as x and y, respectively.

(1) AES: Additive Embeddings Similarity

Given two sentences, x and y, AES between them is computed as follows.

$$\mathrm{AES}(x,y) = \cos\left(\sum_{i=1}^{|x|}\vec{x}_i, \sum_{j=1}^{|y|}\vec{y}_j\right) \tag{4.12}$$

where each sentence is vectorized with the sum of the word embeddings of its component words, $\vec{x}_i$ and $\vec{y}_j$, assuming the additive compositionality [74].

(2) AAS: Average Alignment Similarity

AAS [97] averages the cosine similarities between all pairs of words within given two sentences, x and y, calculated over their embeddings.

$$\mathrm{AAS}(x,y) = \frac{1}{|x||y|}\sum_{i=1}^{|x|}\sum_{j=1}^{|y|}\cos(\vec{x}_i,\vec{y}_j) \tag{4.13}$$

(3) MAS: Maximum Alignment Similarity

AAS inevitably involves noise, as many word pairs are semantically irrelevant to each other. MAS [97] reduces this kind of noise by considering only the best word alignment for each word in one sentence as follows.

$$\mathrm{MAS}(x,y) = \frac{1}{|x|}\sum_{i=1}^{|x|}\max_{j}\cos(\vec{x}_i,\vec{y}_j) \tag{4.14}$$

As MAS is asymmetric, we calculate it for each direction, i.e., $\mathrm{MAS}(x,y)$ and $\mathrm{MAS}(y,x)$, unlike Chapter 3.

(4) HAS: Hungarian Alignment Similarity

AAS and MAS deal with many-to-many and one-to-many word alignments, respectively. On the other hand, HAS [97] is based on one-to-one word alignments.

The task of identifying the best one-to-one word alignments $\mathscr{H}$ is regarded as a problem of bipartite graph matching, where the two sets of vertices respectively comprise words within each sentence x and y, and the weight of a edge between x_i and y_j is given by the cosine similarity calculated over their word embeddings. Given $\mathscr{H}$ identified using the Hungarian algorithm [58], HAS is computed by averaging the similarities between embeddings of the aligned pairs of words.

$$\mathrm{HAS}(x,y) = \frac{1}{|\mathscr{H}|}\sum_{(i,j)\in\mathscr{H}}\cos(\vec{x}_i,\vec{y}_j) \tag{4.15}$$

where $|\mathscr{H}| = \min(|x|,|y|)$, as $\mathscr{H}$ contains only one-to-one word alignments.

(5) WMD: Word Mover's Distance

WMD [60] is a special case of the Earth Mover's Distance [92], which solves the transportation problem of words between two sentences represented by a bipartite graph.[9] Let n be the vocabulary size of the language, WMD is computed as follows.

$$\mathrm{WMD}(x,y) = \min\sum_{u=1}^{n}\sum_{v=1}^{n}\mathscr{A}_{uv}\mathrm{eud}(\vec{x}_u,\vec{y}_v) \tag{4.16}$$

[9]Note that the vertices in the graph represent the word types, unlike the token-based graph for HAS.

$$\text{subject to : } \sum_{v=1}^{n} \mathscr{A}_{uv} = \frac{1}{|x|} freq(x_u, x)$$
$$\sum_{u=1}^{n} \mathscr{A}_{uv} = \frac{1}{|y|} freq(y_v, y)$$

where $\mathscr{A}_{uv}$ is a nonnegative weight matrix representing the flow from a word x_u in x to a word y_v in y, $\text{eud}(\cdot,\cdot)$ the Euclidean distance between two word embeddings, and $freq(\cdot,\cdot)$ the frequency of a word in a sentence.

(6) DWE: Difference of Word Embeddings

We also introduce the difference between sentence embeddings so as to gauge their differences in terms of meaning and simplicity. As the representation of a sentence, we used the averaged word embeddings [1].

$$\text{DWE}(x,y) = \frac{1}{|x|}\sum_{i=1}^{|x|}\vec{x}_i - \frac{1}{|y|}\sum_{j=1}^{|y|}\vec{y}_j \tag{4.17}$$

(7) PAS: Paraphrase Alignment Similarity

PAS [103, 104] is computed based on lexical paraphrases. This feature has been proven useful in the semantic textual similarity task of SemEval-2015 [2].

$$\begin{aligned} \text{PAS}(x,y) &= \frac{\text{PA}(x,y)+\text{PA}(y,x)}{|x|+|y|} \\ \text{PA}(x,y) &= \sum_{i=1}^{|x|} \begin{cases} 1 & \exists j : x_i \Leftrightarrow y_j \in y \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4.18}$$

where $x_i \Leftrightarrow y_j$ holds if and only if the word pair (x_i, y_j) is included in a given paraphrase lexicon.

4.3.2 Settings

The usefulness of the above features was evaluated through an empirical experiment using the QATS dataset [117]. The QATS dataset consists of 505 and 126 sentence

pairs for training and test, respectively, where each pair is evaluated from four different aspects: **G**rammaticality, **M**eaning preservation, **S**implicity, and **O**verall quality. Evaluations are given by one of the three classes: *good*, *ok*, and *bad*.

We used two pre-compiled external resources to compute our features. One is the pre-trained 300-dimensional CBOW model[10] to compute the features based on word embeddings, while the other is PPDB 2.0 [86][11] for PAS.

Each system is evaluated by the three metrics as in the QATS classification task [117]: Accuracy (A), Mean Absolute Error (E) and Weighted F-score (F). To compute Mean Absolute Error, class labels were converted into three equally distant numeric scores retaining their relation, i.e., $good = 1$, $ok = 0.5$, and $bad = 0$.

As the baseline[12], we employed four types of systems from the QATS workshop [117]: two typical baselines and two top-ranked systems. "Majority-class" labels all the sentence pairs with the most frequent class in the training data. "MT-baseline" combines BLEU [84], METEOR [61], TER [96], and WER [63], using a support vector machine (SVM) classifier.

SimpleNets [82] has two different forms of deep neural network architectures: multilayer perceptron (SimpleNets-MLP) and recurrent neural network (SimpleNets-RNN). SimpleNets-MLP uses seven features of each sentence: the number of characters, tokens, and word types, 5-gram language model probabilities estimated on the basis of either SUBTLEX [22], SubIMDB [83], Wikipedia, and Simple Wikipedia [52]. SimpleNets-RNN, which does not require such feature engineering, uses embeddings of word N-grams.

SMH [116] has two types of classifiers: logistic classifier (SMH-IBk/Logistic) and random forest classifier (SMH-RandForest, SMH-RandForest-b). Both are trained relying on the automatic evaluation metrics for MT, such as BLEU, METEOR, and TER, in combination with the QE features for MT [101].

We evaluated our proposed features in the supervised classification fashion as previous work. Specifically, we compared three types of supervised classifiers that had been also used in the above baseline systems: SVM, MLP, and RandForest. Hyperparameters of each system were determined through 5-fold cross validation using the training data, regarding accuracy in terms of overall quality as the objective.

For the SVM classifier, we used the RBF kernel. The trinary classification was

[10] https://code.google.com/archive/p/word2vec/

[11] http://paraphrase.org/

[12] Instead of reimplementing these baseline systems, we excerpted their performance scores from [117].

Table 4.7: Results on QATS classification task. The best scores of each metric are highlighted in bold. Scores other than ours are excerpted from Štajner et al. [117].

System	Grammaticality			Meaning			Simplicity			Overall		
	A↑	E↓	F↑	A↑	E↓	F↑	A↑	E↓	F↑	A↑	E↓	F↑
Majority-class	**76.2**	18.3	65.9	57.9	29.0	42.5	55.6	29.4	39.7	43.7	28.2	26.5
MT-baseline	**76.2**	18.3	65.9	66.7	20.2	62.7	50.8	26.2	48.3	38.1	41.7	37.5
SimpleNets-MLP	74.6	**17.1**	68.8	65.9	21.0	63.5	53.2	27.0	49.8	38.1	32.5	33.7
SimpleNets-RNN ($N=2$)	75.4	18.7	65.5	57.9	27.4	51.3	50.0	27.0	47.5	52.4	25.8	46.1
SimpleNets-RNN ($N=3$)	74.6	19.1	65.1	51.6	28.2	46.6	52.4	25.0	50.0	47.6	27.8	40.8
SMH-IBk/Logistic	70.6	19.4	71.6	**69.1**	20.2	**68.1**	50.0	28.2	51.1	47.6	28.2	47.5
SMH-RandForest	75.4	17.5	**71.8**	65.9	20.6	64.4	52.4	27.8	53.0	44.4	31.8	44.5
SMH-RandForest-b	75.4	18.3	70.0	61.9	23.8	59.7	57.1	25.4	56.4	48.4	29.0	48.6
Best score among the above	**76.2**	**17.1**	**71.8**	**69.1**	**20.2**	**68.1**	57.1	25.0	56.4	52.4	25.8	48.6
Our SVM	**76.2**	18.3	65.9	65.1	22.2	58.3	57.1	27.8	43.9	**57.9**	**23.4**	**57.7**
Our MLP	68.3	24.6	66.9	59.5	25.4	56.4	59.5	23.4	58.2	52.4	25.8	51.9
Our RandForest	**76.2**	18.3	65.9	66.7	23.0	63.2	**63.5**	**21.8**	**59.8**	51.6	26.6	48.3
Our SVM w/ MT-baseline	**76.2**	18.3	65.9	66.7	21.0	63.7	57.1	27.0	46.9	47.6	29.0	46.8
Our MLP w/ MT-baseline	63.5	26.6	63.8	64.3	21.4	62.7	52.4	26.2	53.2	46.0	31.8	45.5
Our RandForest w/ MT-baseline	**76.2**	18.3	65.9	61.9	24.6	57.6	62.7	22.6	56.1	46.0	29.0	43.6

realized by means of the one-versus-the-rest strategy. For a given set of features, we examined all the combinations of hyper-parameters among $C \in \{0.01, 0.1, 1.0\}$ and $\gamma \in \{0.01, 0.1, 1.0\}$; for the full set of features, $C = 1.0$ and $\gamma = 0.1$ were chosen.

As for the MLP classifier, among 1 to 3 layers with all the combinations of dimensionality among $\{100, 200, 300, 400, 500\}$ and "ReLu" for the activation function among {Logistic, tanh, ReLu}, the 2-layer one with 200 $\times$ 200 dimensionality was optimal. We used Adam [54] as the optimizer.

For the RandForest classifier, we examined all the combinations of the following three hyper-parameters: $\{10, 50, 100, 500, 1000\}$ for number of trees, $\{5, 10, 15, 20, \infty\}$ for the maximum depth of each tree, and $\{1, 5, 10, 15, 20\}$ for the minimum number of samples at leaves. The optimal combination for the full set of features was $(500, 15, 1)$.

4.3.3 Results

Experimental results are shown in Table 4.7. The SVM classifier based on our features greatly outperformed the state-of-the-art methods in terms of overall quality. The RandForest classifier somehow achieved the best simplicity scores ever, even though we had optimized the system with respect to the accuracy of overall quality. As we expected, MLP did not beat the other two classifiers, presumably due to the scarcity of

Table 4.8: Ablation analysis on accuracy. Features are in descending order of overall accuracy.

Feature set	C	γ	Grammaticality	Meaning	Simplicity	Overall
ALL	1.0	0.1	76.2	65.1	57.1	57.9
-AES	1.0	0.1	76.2	65.1	57.1	57.1
-MAS(original, simple)	0.1	0.1	76.2	57.9	55.6	56.4
-MAS(simple, original)	1.0	0.1	76.2	64.3	57.1	54.8
-PAS	0.1	0.1	76.2	57.9	55.6	53.2
-DWE	0.01	1.0	76.2	57.9	55.6	51.6
-WMD	0.01	0.1	76.2	57.9	55.6	46.8
-AAS	0.1	0.1	76.2	57.9	55.6	45.2
-HAS	0.01	0.01	76.2	57.9	55.6	35.7

Table 4.9: An example of word alignment. Differences between the original and simplified versions are presented in bold. This is a sentence pair from *good* class on overall quality. HAS using word-level similarity reaches 0.85, while BLEU is 0.54.

Original	**While** historians **concur** that the result **itself** was not **manipulated**, the voting process was neither free nor secret.
Simple	**Most** historians **agree** that the result was not **fixed**, **but** the voting process was neither free nor secret.
Hungarian Alignment	(while, but), (concur, agree), (itself, most), (manipulated, fixed), and identical word pairs.

the training data.

The bottom three rows reveal that the performance in terms of overall quality was deteriorated when MT-baseline features were incorporated on top of our feature set. This suggests that word embeddings are superior to surface-level processing in finding corresponding words within sentence pairs.

Focusing on the overall quality, we conducted an ablation analysis of the SVM classifier. The analysis revealed, as shown in Table 4.8, that HAS, AAS, and WMD were the most important features. This can be explained by the role of word alignments during the computation. Since MT metrics, such as BLEU, rely only on surface-level matches, they are insensitive to meaning-preserving rewritings from original sentence to simple one. On the other hand, as exemplified in Table 4.9, HAS and some other features can detect the linkages between complex words and their simpler counterparts. As a result of properly capturing the alignments between such lexical paraphrases, our

Table 4.10: Correlation between each feature and the difference of sentence length and the manually-labeled quality. Note that DWE cannot be included, as it is not a scalar value but the differential vector between original and simplified sentences.

Feature	r_{length}	r_{label}
AES	-0.661	0.185
AAS	-0.335	0.318
MAS(original, simple)	-0.817	0.226
MAS(simple, original)	0.092	-0.090
HAS	0.061	-0.050
WMD	0.788	-0.215
PAS	-0.120	-0.039

system successfully classified this sentence into *good* in terms of overall quality.

We expected that AAS could yield noise, as it involves irrelevant pairs of words, but in fact, it contributed to the QATS task. We speculate that it helps to evaluate the appropriateness of substituting a word to other one considering the semantic matching with the given context, as in lexical simplification [16] and lexical substitution [72, 90, 8].

The contribution of WMD was expected as it was proven effective in the sentence alignment task of English Wikipedia and Simple English Wikipedia.

Table 4.10 shows that some of our semantic similarity features are also strongly biased by the length difference between original and simple sentences, as MT metrics (cf. Table 4.6). Nonetheless, HAS was not biased by the length difference almost at all, and AAS and WMD highly correlated with the manually-labeled quality.

4.3.4 Relationship between Word Embeddings and Word Difficulty

We will examine why simplicity was successfully evaluated by using features based on word embeddings. We train word embeddings on English Wikipedia[13] for all combinations of the following conditions:

- Number of dimension: 100, 200, 500
- Window size: 1, 3, 5, 10
- Algorithm: CBOW [73], SGNS [73], GloVe [87]

[13] https://dumps.wikimedia.org/enwiki/20171201/

Table 4.11: CBOW

Window	100d	200d	500d
1	0.215	0.195	0.212
3	0.123	0.121	0.129
5	0.156	0.128	0.096
10	0.112	0.130	0.120

Table 4.12: SGNS

Window	100d	200d	500d
1	0.219	0.222	0.191
3	0.259	0.303	0.225
5	0.271	0.261	0.252
10	0.293	0.275	0.281

Table 4.13: GloVe

Window	100d	200d	500d
1	0.329	0.321	0.293
3	0.425	0.412	0.421
5	0.450	0.433	0.411
10	0.444	0.460	0.476

We evaluate the Pearson correlation coefficient between these word embeddings and the manually given word difficulty. For evaluation, we use the dataset of SemEval-2016 Complex Word Identification task[14] [81]. This is a corpus in which 20 annotators give labels of either complex (1) or simple (0) for each 2,237 target words. In this work, we average the labels of all annotators and define word difficulty in the range of $[0.00, 1.00]$ for each target words.

Tables 4.11, 4.12 and 4.13 show the maximum value of the Pearson correlation coefficient between each dimension of word embeddings and word difficulty. In the CBOW model of word2vec [73], dimensions having a week correlation ($r > 0.2$) with word difficulty appear when training word embeddings using window size 1, that is, words adjacent to the target word. In the SGNS model of word2vec [73], dimensions having a week correlation with word difficulty appear in many settings. In GloVe [87], dimensions having a moderate correlation ($r > 0.4$) with word difficulty appear in many settings. Unlike CBOW and SGNS, GloVe explicitly considers word frequency in the corpus. As shown in SemEval-2012 English Lexical Simplification task [99], word frequency greatly affects word difficulty, so we believe that GloVe embeddings showed a stronger correlation with word difficulty.

From the above experimental results, it seems that dimensions that govern word difficulty appear in word embeddings. Therefore, in the experiment of Table 4.7, word difficulty can be taken into consideration by using features based on word embeddings, and simplicity can be better evaluated than in previous works.

[14] http://alt.qcri.org/semeval2016/task11/

Chapter 5

Final Remarks

5.1. Conclusion

In this thesis, we worked on text simplification in lexical substitution approach and monolingual translation approach for languages that cannot use simplified correct. For lexical substitution approach, we proposed three types of candidate acquisition, four types of meaning preservation filtering, four types of simplicity filtering, and three types of grammaticality ranking methods. Moreover, we built an evaluation dataset for Japanese lexical simplification that solves the problems of previous evaluation datasets in English [99, 12, 44, 80]. This dataset enabled automatic evaluation for Japanese lexical simplification systems. Experimental results showed that the proposed method (synonym dictionaries constructed manually + language model ranking) outperformed the previous language-independent unsupervised method [34]. We also confirmed that the synonym dictionary constructed automatically can acquire more simplification rules than other methods and is the most promising from the viewpoint of oracle accuracy.

For monolingual translation approach, we proposed a method to construct pseudo-parallel corpus from a raw corpus using readability assessment and sentence alignment for languages that cannot use parallel corpora. First, we showed the usefulness of the sentence similarity based on alignment between word embeddings in the alignment task of complex and simple sentences [46]. Next, we confirmed that the pseudo-parallel corpus constructed from a raw corpus is effective for training PBSMT model as much as using the existing parallel corpus for text simplification [128, 27, 46]. Since many languages other than English cannot use a simplified large-scale corpus, this work opens the door to text simplification for many other languages.

Along with the above main contributions, we further improve text simplification through English experiments. First, we tackle the paraphrase acquisition task which is important for improving the lexical substitution approach. Next, we address the sentence similarity task which is important for improving the monolingual translation approach. Finally, we work on the quality estimation task for improving the automatic evaluation metrics of text simplification. With all these tasks, we achieved state-of-the-art performance.

For paraphrase acquisition, we proposed a paraphrasability score that complements the paraphrasability from monolingual and bilingual corpora. This work gives a novel interpretation that bilingual pivoting [10], the de facto standard method for paraphrase acquisition, is an unsmoothed version of weighted pointwise mutual information.

Moreover, we proposed a domain adaptation method for sentence similarity measurement. This is an updating method general word similarities with word similarities specialised to a given corpus. Experimental results showed that the proposed iterative method is significantly better than the non-iterative counterparts.

Finally, we proposed a quality estimation method for text simplification. This work showed that sentence similarities based on alignment between word embeddings are useful for quality estimation of text simplification, and greatly improved the state-of-the-art methods [82, 116].

5.2. Future Work

In this thesis, we proposed a novel lexical substitution approach and a monolingual translation approach for the purpose of multilingualization of text simplification. Future works are listed below.

To consider phrases in lexical substitution approach

In previous lexical substitution approaches, including our work, we have only targeted words. In fact, it may be difficult to explain with simple words, but it can be paraphrases with simple phrases. The evaluation dataset we constructed contains the phrasal substitution. In word-to-word substitution, oracle accuracy is 0.211, but future expansioin is expeted by extending to phrase.

To generate monolingual parallel corpus automatically

Automatic construction of monolingual parallel corpus using back-translation

and roundtrip-translation [93, 107, 47] is expected by advancing machine translation [106, 9, 41]. If a monolingual parallel corpus can be constructed on a large-scale and with high-quality, it can be used for a lexical substitution approach using paraphrase acquisition methods. In addition, similar to our work, parallel corpus for text simplification can be constructed by combining with readability assessment.

Bibliography

[1] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, and Y. Goldberg. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. In *International Conference on Learning Representations*, pp. 1–13, 2017.

[2] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, G. Rigau, and J. Wiebe. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pp. 252–263, 2015.

[3] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, pp. 81–91, 2014.

[4] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. Wiebe. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 497–511, 2016.

[5] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics*, pp. 385–393, 2012.

[6] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo. *SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics*, pp. 32–43, 2013.

[7] S. Amano and K. Kondo. On the NTT Psycholinguistic Databases "Lexical-Properties of Japanese". *Journal of the Phonetic Society*, 4(2):44–50, 2000.

[8] M. Apidianaki. Vector-space models for PPDB paraphrase ranking in context. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2028–2034, 2016.

[9] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*, pp. 1–15, 2015.

[10] C. Bannard and C. Callison-Burch. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 597–604, 2005.

[11] J. D. Belder and M.-F. Moens. Text Simplification for Children. In *Proceedings of the SIGIR Workshop on Accessible Search Systems*, pp. 19–26, 2010.

[12] J. D. Belder and M.-F. Moens. A Dataset for the Evaluation of Lexical Simplification. In *Proceedings of the 13th International Conference on Computational Linguistics and Intelligent Text Processing*, pp. 426–437, 2012.

[13] R. Bhagat and D. Ravichandran. Large Scale Acquisition of Paraphrases for Learning Surface Patterns. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 674–682, 2008.

[14] S. Biggins, S. Mohammed, S. Oakley, L. Stringer, M. Stevenson, and J. Priess. University_Of_Sheffield: Two Approaches to Semantic Text Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics*, pp. 655–661, 2012.

[15] J. Bingel and A. Søgaard. Text Simplification as Tree Labeling. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 337–343, 2016.

[16] O. Biran, S. Brody, and N. Elhadad. Putting it Simply: a Context-Aware Approach to Lexical Simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 496–501, 2011.

[17] O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pp. 1–44, 2013.

[18] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 12–58, 2014.

[19] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, C. Monz, M. Negri, M. Post, R. Rubino, L. Specia, and M. Turchi. Findings of the 2017 Conference on Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, pp. 169–214, 2017.

[20] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Neveol, M. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor, and M. Zampieri. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pp. 131–198, 2016.

[21] O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, M. Huck, C. Hokamp, P. Koehn, V. Logacheva, C. Monz, M. Negri, M. Post, C. Scarton, L. Specia, and M. Turchi. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 1–46, 2015.

[22] M. Brysbaert and B. New. Moving beyond Kučera and Francis: A Critical Evaluation of Current Word Frequency Norms and the Introduction of a New and Improved Word Frequency Measure for American English. *Behavior Research Methods*, 41(4):977–990, 2009.

[23] C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pp. 10–51, 2012.

[24] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, pp. 1–14, 2017.

[25] T. P. Chan, C. Callison-Burch, and B. V. Durme. Reranking Bilingually Extracted Paraphrases Using Monolingual Distributional Similarity. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pp. 33–42, 2011.

[26] W. Coster and D. Kauchak. Learning to Simplify Sentences Using Wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pp. 1–9, 2011.

[27] W. Coster and D. 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.

[28] S. Devlin and J. Tait. The use of a Psycholinguistic Database in the Simplification of Text for Aphasic Readers. *Linguistic Databases*, pp. 161–173, 1998.

[29] S. Devlin and G. Unthank. Helping Aphasic People Process Online Information. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 225–226, 2006.

[30] R. J. Evans. Comparing methods for the syntactic simplification of sentences in information extraction. *Literary and Linguistic Computing*, 26(4):371–388, 2011.

[31] S. Evert. *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. PhD thesis, University of Stuttgart, 2005.

[32] R. Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32:221–233, 1948.

[33] J. Ganitkevitch, B. V. Durme, and C. Callison-Burch. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 758–764, 2013.

[34] G. Glavaš and S. Štajner. Simplifying Lexical Simplification: Do We Need Simplified Corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 63–68, 2015.

[35] I. Goto, H. Tanaka, and T. Kumano. Japanese News Simplification: Task Design, Data Set Construction, and Analysis of Simplified Text. In *Proceedings of MT Summit XV*, pp. 17–31, 2015.

[36] W. Guo and M. Diab. Modeling Sentences in the Latent Space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 864–872, 2012.

[37] L. Han, A. Kashyap, T. Finin, J. Mayfield, and J. Weese. UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Second Joint Conference on Lexical and Computational Semantics*, pp. 44–52, 2013.

[38] Z. S. Harris. Distributional Structure. *Word*, 10(23):146–162, 1954.

[39] C. Hashimoto, K. Torisawa, K. Kuroda, M. Murata, and J. Kazama. Large-Scale Verb Entailment Acquisition from the Web. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 1172–1181, 2009.

[40] C. Hashimoto, K. Torisawa, S. D. Saeger, J. Kazama, and S. Kurohashi. Extracting Paraphrases from Definition Sentences on the Web. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 1087–1097, 2011.

[41] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma. Dual Learning for Machine Translation. In *Advances in Neural Information Processing Systems 29*, pp. 820–828, 2016.

[42] K. Heafield. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pp. 187–197, 2011.

[43] F. Hill, K. Cho, and A. Korhonen. Learning Distributed Representations of Sentences from Unlabelled Data. In *Proceedings of the 2016 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1367–1377, 2016.

[44] C. Horn, C. Manduca, and D. Kauchak. Learning a Lexical Simplifier Using Wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 458–463, 2014.

[45] B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Advances in Neural Information Processing Systems 27*, pp. 2042–2050, 2014.

[46] W. Hwang, H. Hajishirzi, M. Ostendorf, and W. 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.

[47] A. Imankulova, T. Sato, and M. Komachi. Improving Low-Resource Neural Machine Translation with Filtered Pseudo-Parallel Corpus. In *Proceedings of the 4th Workshop on Asian Translation*, pp. 70–78, 2017.

[48] M. Imono, E. Yoshimura, S. Tsuchiya, and H. Watabe. Proposal of a Method to Convert Difficult Words in Newspaper Articles to Plain Expressions. *Journal of Natural Language Processing*, 20(2):105–132, 2013.

[49] H. Isahara, F. Bond, K. Uchimoto, M. Utiyama, and K. Kanzaki. Development of the Japanese WordNet. In *Proceedings of the Sixth conference on International Language Resources and Evaluation*, pp. 2420–2423, 2008.

[50] Y. Ji and J. Eisenstein. Discriminative Improvements to Distributional Sentence Similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 891–896, 2013.

[51] N. Kaji, D. Kawahara, S. Kurohashi, and S. Sato. Verb Paraphrase based on Case Frame Alignment. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 215–222, 2002.

[52] D. Kauchak. Improving Text Simplification Language Modeling Using Unsimplified Text Data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 1537–1546, 2013.

[53] T. Kenter and M. de Rijke. Short Text Similarity with Word Embeddings . In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 1411–1420, 2015.

[54] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, pp. 1–15, 2015.

[55] P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Machine Translation Summit*, pp. 79–86, 2005.

[56] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pp. 177–180, 2007.

[57] T. Kudo, K. Yamamoto, and Y. 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.

[58] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[59] S. Kurohashi, T. Nakamura, Y. Matsumoto, and M. Nagao. Improvements of Japanese Morphological Analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language*, pp. 22–28, 1994.

[60] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. From Word Embeddings To Document Distances. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 957–966, 2015.

[61] A. Lavie and M. Denkowski. The METEOR Metric for Automatic Evaluation of Machine Translation. *Machine Translation*, 23(2-3):105–115, 2009.

[62] Q. Le and T. Mikolov. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1188–1196, 2014.

[63] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[64] O. Levy and Y. Goldberg. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 302–308, 2014.

[65] O. Levy and Y. Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems 27*, pp. 2177–2185, 2014.

[66] D. Lin and P. Pantel. Discovery of Inference Rules for Question Answering. *Natural Language Engineering*, 7(4):343–360, 2001.

[67] K. Maekawa, M. Yamazaki, T. Maruyama, M. Yamaguchi, H. Ogura, W. Kashino, T. Ogiso, H. Koiso, and Y. Den. Design, Compilation, and Preliminary Analyses of Balanced Corpus of Contemporary Written Japanese. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pp. 1483–1486, 2010.

[68] B. Marie and A. Fujita. Efficient Extraction of Pseudo-Parallel Sentences from Raw Monolingual Data Using Word Embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 392–398, 2017.

[69] Y. Marton. Distributional Phrasal Paraphrase Generation for Statistical Machine Translation. *ACM Transactions on Intelligent Systems and Technology*, 4(3):1–32, 2013.

[70] T. Matsui, Y. Baba, T. Kamishima, and H. Kashima. Crowdordering. In *Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 336–347, 2014.

[71] D. McCarthy and R. Navigli. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pp. 48–53, 2007.

[72] O. Melamud, O. Levy, and I. Dagan. A Simple Word Embedding Model for Lexical Substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 1–7, 2015.

[73] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at the International Conference on Learning Representations*, pp. 1–12, 2013.

[74] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and Their Compositionality. In *Advances in Neural Information Processing Systems 26*, pp. 3111–3119, 2013.

[75] G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[76] M. Mizukami, G. Neubig, S. Sakti, T. Toda, and S. Nakamura. Building a Free, General-Domain Paraphrase Database for Japanese. In *Proceedings of the 17th Oriental COCOSDA Conference*, pp. 129–133, 2014.

[77] S. M. Mohammad, B. J. Dorr, G. Hirst, and P. D. Turney. Computing Lexical Contrast. *Computational Linguistics*, 39(3):555–590, 2013.

[78] Y. Niwa and Y. Nitta. CO-OCCURRENCE VECTORS FROM CORPORA VS. DISTANCE VECTORS FROM DICTIONARIES. In *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 304–309, 1994.

[79] F. J. Och and H. Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 2003.

[80] G. H. Paetzold and L. Specia. Benchmarking Lexical Simplification Systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pp. 3074–3080, 2016.

[81] G. H. Paetzold and L. Specia. SemEval 2016 Task 11: Complex Word Identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 560–569, 2016.

[82] G. H. Paetzold and L. Specia. SimpleNets: Evaluating Simplifiers with Resource-Light Neural Networks. In *Proceedings of the LREC 2016 Workshop & Shared Task on Quality Assessment for Text Simplification*, pp. 42–46, 2016.

[83] G. H. Paetzold and L. Specia. Unsupervised Lexical Simplification for Non-Native Speakers. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 3761–3767, 2016.

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

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

[86] E. Pavlick, P. Rastogi, J. Ganitkevitch, B. V. Durme, and C. Callison-Burch. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 425–430, 2015.

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

[88] S. E. Petersen and M. Ostendorf. Text Simplification for Language Learners: A Corpus Analysis. In *Proceedings of the Speech and Language Technology in Education Workshop*, pp. 69–72, 2007.

[89] D. Ravichandran, P. Pantel, and E. Hovy. Randomized Algorithms and NLP: Using Locality Sensitive Hash Functions for High Speed Noun Clustering. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 622–629, 2005.

[90] S. Roller and K. Erk. PIC a Different Word: A Simple Model for Lexical Substitution in Context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1121–1126, 2016.

[91] S. Ross, G. Gordon, and D. Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 627–635, 2011.

[92] Y. Rubner, C. Tomasi, and L. J. Guibas. A Metric for Distributions with Applications to Image Databases. In *Proceedings of the Sixth International Conference on Computer Vision*, pp. 59–66, 1998.

[93] R. Sennrich, B. Haddow, and A. Birch. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 86–96, 2016.

[94] M. Shardlow. A Survey of Automated Text Simplification. *International Journal of Advanced Computer Science and Applications, Special Issue on Natural Language Processing*, pp. 58–70, 2014.

[95] S. Sharoff. Open-source corpora: Using the net to fish for linguistic data. *International Journal of Corpus Linguistics*, 11(4):435–462, 2006.

[96] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pp. 1–9, 2006.

[97] Y. Song and D. 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.

[98] L. 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.

[99] L. Specia, S. K. Jauhar, and R. Mihalcea. SemEval-2012 Task 1: English Lexical Simplification. In *Proceedings of the *SEM 2012: The First Joint Conference on Lexical and Computational Semantics*, pp. 347–355, 2012.

[100] L. Specia, D. Raj, and M. Turchi. Machine Translation Evaluation versus Quality Estimation. *Machine Translation*, 24(1):39–50, 2010.

[101] L. Specia, K. Shah, J. G. de Souza, and T. Cohn. QuEst - A translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 79–84, 2013.

[102] L. Specia, M. Turchi, N. Cancedda, M. Dymetman, and N. Cristianini. Estimating the Sentence-Level Quality of Machine Translation Systems. In *13th Annual Conference of the European Association for Machine Translation*, pp. 28–37, 2009.

[103] M. A. Sultan, S. Bethard, and T. Sumner. Back to Basics for Monolingual Alignment: Exploiting Word Similarity and Contextual Evidence. *Transactions of the Association for Computational Linguistics*, 2:219–230, 2014.

[104] M. A. Sultan, S. Bethard, and T. Sumner. DLS@CU: Sentence Similarity from Word Alignment and Semantic Vector Composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pp. 148–153, 2015.

[105] Y. Sunakawa, J. ho Lee, and M. Takahara. The Construction of a Database to Support the Compilation of Japanese Learners Dictionaries. *Acta Linguistica Asiatica*, 2(2):97–115, 2012.

[106] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, pp. 3104–3112, 2014.

[107] Y. Suzuki, T. Kajiwara, and M. Komachi. Building a Non-Trivial Paraphrase Corpus Using Multiple Machine Translation Systems. In *Proceedings of ACL 2017 Student Research Workshop*, pp. 36–42, 2017.

[108] I. Szpektor and I. Dagan. Learning Entailment Rules for Unary Templates. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 849–856, 2008.

[109] G. Tsatsaronis, I. Varlamis, and M. Vazirgiannis. Text Relatedness Based on a Word Thesaurus. *Journal of Artificial Intelligence Research*, 37:1–39, 2010.

[110] P. D. Turney and P. Pantel. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

[111] S. Štajner. Translating sentences from 'original' to 'simplified' Spanish. *Procesamiento del Lenguaje Natural*, 53:61–68, 2014.

[112] S. Štajner, H. Bechara, and H. Saggion. A Deeper Exploration of the Standard PB-SMT Approach to Text Simplification and its Evaluation. In *Proceedings*

of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 823–828, 2015.

[113] S. Štajner, I. Calixto, and H. Saggion. Automatic Text Simplification for Spanish: Comparative Evaluation of Various Simplification Strategies. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pp. 618–626, 2015.

[114] S. Štajner, R. Mitkov, and H. Saggion. One Step Closer to Automatic Evaluation of Text Simplification Systems. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pp. 1–10, 2014.

[115] S. Štajner and M. Popović. Can Text Simplification Help Machine Translation? *Baltic Journal of Modern Computing*, 4(2):230–242, 2016.

[116] S. Štajner, M. Popović, and H. Béchara. Quality Estimation for Text Simplification. In *LREC 2016 Workshop & Shared Task on Quality Assessment for Text Simplification*, pp. 15–21, 2016.

[117] S. Štajner, M. Popović, H. Saggion, L. Specia, and M. Fishel. Shared Task on Quality Assessment for Text Simplification. In *LREC 2016 Workshop & Shared Task on Quality Assessment for Text Simplification*, pp. 22–31, 2016.

[118] S. Štajner and H. Saggion. Translating from Original to Simplified Sentences using Moses: When does it Actually Work? In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pp. 611–617, 2015.

[119] J. Weeds and D. Weir. A General Framework for Distributional Similarity. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pp. 81–88, 2003.

[120] S. Wubben, A. van den Bosch, and E. Krahmer. Sentence Simplification by Monolingual Machine Translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 1015–1024, 2012.

[121] W. Xu, C. Callison-Burch, and C. Napoles. Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics*, 3:283–297, 2015.

[122] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch. Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415, 2016.

[123] K. Yamamoto and K. Yoshikura. Manual Construction of Lexical Paraphrase Dictionary of Japanese Verbs, Adjectives, and Adverbs. In *Proceedings of the 19th Annual Meeting of Association for Natural Language Processing*, pp. 276–279, 2013.

[124] M. Yatskar, B. Pang, C. Danescu-Niculescu-Mizil, and L. Lee. For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In *Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 365–368, 2010.

[125] D. Yogatama and N. A. Smith. Making the Most of Bag of Words: Sentence Regularization with Alternating Direction Method of Multipliers. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 656–664, 2014.

[126] M. Yu and M. Dredze. Improving Lexical Embeddings with Semantic Knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 545–550, 2014.

[127] X. Zhang and M. Lapata. Sentence Simplification with Deep Reinforcement Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 595–605, 2017.

[128] Z. Zhu, D. Bernhard, and I. Gurevych. A Monolingual Tree-based Translation Model for Sentence Simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 1353–1361, 2010.

Appendix A

Related Publications

Journal Papers

1. Tomoyuki Kajiwara, Mamoru Komachi. **Text Simplification without Simplified Corpora.** Journal of Natural Language Processing, Vol.25, No.2, 2018. (to appear)

2. Tomoyuki Kajiwara, Danushka Bollegala, Yuichi Yoshida, Ken-ichi Kawarabayashi. **An Iterative Approach for the Global Estimation of Sentence Similarity.** PLOS ONE, Vol.12, No.9, pp.1-15, 2017.

3. Tomoyuki Kajiwara, Kazuhide Yamamoto. **Japanese Lexical Simplification for Children Using Definition Statements.** Journal of Information Processing Society of Japan, Vol.56, No.3, pp.983-992. 2015.

Refereed Conference Papers

1. Tomoyuki Kajiwara, Mamoru Komachi, Daichi Mochihashi. **MIPA: Mutual Information Based Paraphrase Acquisition via Bilingual Pivoting.** In Proceedings of the 8th International Joint Conference on Natural Language Processing, pp.80–89. 2017.

2. Tomoyuki Kajiwara, Atsushi Fujita. **Semantic Features Based on Word Alignments for Estimating Quality of Text Simplification.** In Proceedings of the 8th International Joint Conference on Natural Language Processing, pp.109–115. 2017.

3. Tomoyuki Kajiwara, Mamoru Komachi. **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. 2016.

4. Tomonori Kodaira, Tomoyuki Kajiwara, Mamoru Komachi. **Controlled and Balanced Dataset for Japanese Lexical Simplification.** In Proceedings of the ACL 2016 Student Research Workshop, pp.1-7. 2016.

5. Tomoyuki Kajiwara, Kazuhide Yamamoto. **Evaluation Dataset and System for Japanese Lexical Simplification.** In Proceedings of the ACL-IJCNLP 2015 Student Research Workshop, pp.35-40. 2015.

6. Tomoyuki Kajiwara, Hiroshi Matsumoto, Kazuhide Yamamoto. **Selecting Proper Lexical Paraphrase for Children.** In Proceedings of the 25th Conference on Computational Linguistics and Speech Processing, pp.769-772. 2013.

Domestic Conference Papers

1. Tomoyuki Kajiwara, Mamoru Komachi, Daichi Mochihashi. **Mutual Information Based Paraphrase Acquisition via Bilingual Pivoting（Bilingual Pivoting による言い換え獲得の相互情報量に基づく一般化）.** The 231st Information Processing Society of Japan Special Interest Group of Natural Language Processing, Vol.2017–NL–231, No.21, pp.1–8. 2017.

2. Tomoyuki Kajiwara, Mamoru Komachi. **Simple PPDB: Japanese.** In Proceedings of the 23rd annual meeting of the Association for Natural Language Processing, pp.529–532. 2017.

3. Tomoyuki Kajiwara, Mamoru Komachi. **Text Simplification without Simplified Corpora**（平易なコーパスを用いないテキスト平易化のための単言語パラレルコーパスの構築）**.** The 229th Information Processing Society of Japan Special Interest Group of Natural Language Processing, Vol.2016–NL–229, No.13, pp.1–8. 2016.

4. Tomoyuki Kajiwara, Mamoru Komachi. **Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment**

between Word Embeddings（単語分散表現のアライメントに基づく文間類似度を用いたテキスト平易化のための単言語パラレルコーパスの構築）**.** The 11st Symposium of Young Researcher Association for Natural Language Processing Studies, P31, 2016.

5. Tomoyuki Kajiwara, Mamoru Komachi. **Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment between Word Embeddings**（単語分散表現のアライメントに基づく文間類似度を用いたテキスト平易化のための単言語パラレルコーパスの構築）**.** The 227th Information Processing Society of Japan Special Interest Group of Natural Language Processing, Vol.2016–NL–227, No.12, pp.1–8. 2016.

6. Tomonori Kodaira, Tomoyuki Kajiwara, Mamoru Komachi. **Controlled and Balanced Dataset for Japanese Lexical Simplification**（均衡コーパスを用いた日本語語彙平易化データセットの構築）**.** In Proceedings of the 22nd annual meeting of the Association for Natural Language Processing, pp.258–261. 2016.

7. Tomonori Kodaira, Tomoyuki Kajiwara, Mamoru Komachi. **Improving Evaluation Dataset for Japanese Lexical Simplification**（語彙平易化システムの評価のためのデータセットの改良）**.** The 10th Symposium of Young Researcher Association for Natural Language Processing Studies, P14. 2015.

8. Tomoyuki Kajiwara, Mamoru Komachi. **Guideline for Text Simplification Corpus**（テキスト平易化コーパスの構築指針）**.** The 10th Symposium of Young Researcher Association for Natural Language Processing Studies, P05. 2015.

9. Tomoyuki Kajiwara, Kazuhide Yamamoto. **Evaluation Dataset for Japanese Lexical Simplification**（日本語の語彙平易化評価セットの構築）**.** In Proceedings of the 21st annual meeting of the Association for Natural Language Processing, pp.501–504. 2015.

10. Tomoyuki Kajiwara, Kazuhide Yamamoto. **System for Japanese Lexical Simplification**（日本語の語彙平易化システムの構築）**.** In Proceedings of the 77th National Convention of Information Processing Society of Japan, pp.167–168. 2015.

11. Tomoyuki Kajiwara, Kazuhide Yamamoto. **Qualitative Evaluation of Available Japanese Resources for Lexical Paraphrasing**（日本語の語彙的換言知

識の質的評価）. The Institute of Electronics, Information and Communication Engineers Technical Report. Natural language understanding and models of communication, Vol.114, No.366, pp.43–48. 2014.

12. Tomoyuki Kajiwara, Kazuhide Yamamoto. **Selecting Proper Lexical Paraphrase for Children**（小学生の読解支援に向けた語釈文から語彙的換言を選択する手法）. The 8th Symposium of Young Researcher Association for Natural Language Processing Studies, P23. 2013.

13. Tomoyuki Kajiwara, Kazuhide Yamamoto. **Lexical Simplification Using Multiple Paraphrase Lexicons**（小学生の読解支援に向けた複数の換言知識を併用した語彙平易化と評価）. In Proceedings of the 19th annual meeting of the Association for Natural Language Processing, pp.272–275. 2013.

14. Tomoyuki Kajiwara, Kazuhide Yamamoto. **Lexical Simplification for Children Using Definition Statements**（小学生の読解支援に向けた語釈文による換言）. The 7th Symposium of Young Researcher Association for Natural Language Processing Studies, P01. 2012.

Awards

1. **Student Incentive Award**, The 231st Information Processing Society of Japan Special Interest Group of Natural Language Processing.

2. **Outstanding Research Award**, The 229th Information Processing Society of Japan Special Interest Group of Natural Language Processing.

3. **Student Incentive Award**, The 77th National Convention of Information Processing Society of Japan.

4. **Incentive Award**, The 7th Symposium of Young Researcher Association for Natural Language Processing Studies.