BERT-Based Simplification of Japanese Sentence-Ending Predicates in Descriptive Text

Taichi Kato **Rei Miyata** **Satoshi Sato**
Graduate School of Engineering
Nagoya University
{kato.taichi,miyata.rei,sato.satoshi}@{j,c,d}.mbox.nagoya-u.ac.jp

Abstract

Japanese sentence-ending predicates intricately combine content words and functional elements, such as aspect, modality, and honorifics; this can often hinder the understanding of language learners and children. Conventional lexical simplification methods, which replace difficult target words with simpler synonyms acquired from lexical resources in a word-by-word manner, are not always suitable for the simplification of such Japanese predicates. Given this situation, we propose a BERT-based simplification method, the core feature of which is the high ability to substitute the whole predicates with simple ones while maintaining their core meanings in the context by utilizing pre-trained masked language models. Experimental results showed that our proposed methods consistently outperformed the conventional thesaurus-based method by a wide margin. Furthermore, we investigated in detail the effectiveness of the average token embedding and dropout, and the remaining errors of our BERT-based methods.

1 Introduction

Lexical simplification (LS) is the process of replacing complex words into easy words or short phrases without changing their meaning to simplify text for an easier understanding (De Belder and Moens, 2010; Paetzold and Specia, 2016). LS has also shown to be effective in improving the downstream natural language processing tasks, such as machine translation (Štajner and Popovic, 2016). Popular LS methods use a paraphrase lexicon that pairs complex and easy words, built from various language resources (Pavlick and Callison-Burch, 2016; Nishihara and Kajiwara, 2020).

However, while the conventional LS methods are widely applicable to various sentences, they are not necessarily effective for tasks in which relatively longer linguistic spans should be treated comprehensively. The simplification of Japanese sentence-ending predicates is one such task. These predicates comprise multiple parts, i.e., a particle, content part (mainly verbs, nouns and adjectives) and functional part (mainly postpositional particles and auxiliary verbs),[1] as shown in Figure 1. Because the functional part tends to combine aspect, modality, and honorifics, the understanding of language learners and children may be hindered.

Figure 1 also shows an example of a manual simplification of the predicate; whole elements in the phrase are jointly rewritten. From this example, it is discovered that although the original meaning of the main verb "*haichisuru* (be arranged)" is reduced to the simpler general word "*aru* (there are)," the core meaning of the sentence can still be successfully conveyed. We assume that this interesting fact may generally hold for a certain type of text, namely, descriptive text, which explains the attributes and states of objects and situations (Robert and Wolfgang, 1981). In descriptive sentences, core information is chiefly encoded in the arguments and their modifiers rather than the predicates. In fact, in many cases, if predicates are omitted from descriptive sentences, humans can reasonably complement alternative phrases by referring only to the remaining part. This observation motivated us to replicate these human intuitions by utilizing masked language models, which can predict the omitted tokens in a sentence.

Therefore, in this paper, we propose a BERT-based simplification method for Japanese sentence-ending predicates. Our key idea is to use a masked

[1] In this study, we extended the definition of the Japanese predicate by Kawahara et al. (2002) to include a preceding particle, which is useful for the simplification task. As Japanese is a head-final language, the main predicate of a sentence is usually located at the end of the sentence.

Proceedings of The 13th International Conference on Natural Language Generation, pages 242–251,
Dublin, Ireland, 15-18 December, 2020.

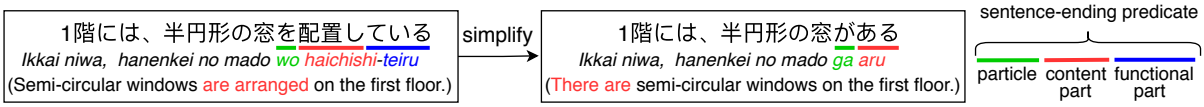

Figure 1: Example of Japanese sentence-ending predicate and its simplified version.

language model to sequentially restore a predicate that is masked entirely. Our method also incorporate the mechanisms to *partially* encode the meaning of original predicates using average token embedding and dropout to generate various simple candidate phrases that are moderately related to the original one. We experimentally evaluated the performance of the proposed method by comparing it with a thesaurus-based LS method; the high ability of our method was revealed with detailed discussions and analyses.

2 Related Work

Automatic text simplification (ATS) is a task of rewriting a complex text as simpler text while maintaining its meaning. Inspired by the machine translation (MT) task, recent ATS approaches regard the text simplification as a monolingual MT task (Wubben et al., 2012; Nisioi et al., 2017). Although sequence-to-sequence neural MT has contributed to better simplification performance, a large amount of parallel data that pairs original-to-simplified text is required. In English, various types of such datasets are available, such as the PWKP dataset (Zhu et al., 2010), which was constructed by automatically aligning sentences from English Wikipedia and Simple English Wikipedia.[2] Xu et al. (2015) introduced the Newsela corpus, which comprises news articles rewritten by professional editors.

Several attempts have also been made to develop datasets for Japanese simplification, such as one built by using crowdsourcing (Katsuta and Yamamoto, 2018). However, compared with the resources available in English, they are not sufficiently large to build robust models. Therefore, for languages with limited resources of aligned corpora for simplification, even including Japanese, NMT-based ATS methods are not necessarily feasible.

LS is a subtask of ATS that aims to substitute complex words with easy words. One of the most popular LS systems uses a simplified synonym lexicon extracted from WordNet and other resources (S. Rebecca and Sven, 2012). Simple-PPDB (Pavlick and Callison-Burch, 2016) is a large-scale complex-to-simple paraphrase dataset built from PPDB (Pavlick et al., 2015). Glavaš and Štajner (2015) used word embeddings to obtain synonyms of the target word. A recent study (Zhou et al., 2019) proposed a BERT-based LS method which produces simpler words that are both semantically and contextually similar to the target word. However, these LS methods adopt the word-by-word substitution approach, and in principle, do not take longer units of sentences as input, such as Japanese sentence-ending predicates.

To manage longer text spans without recourse to sequence-to-sequence generation methods, leveraging the general-purpose monolingual models that are trained on large data might be effective. In recent years, various types of pre-trained models have been proposed and trained for many languages (Qiu et al., 2020). For Japanese, several models are available, such as BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018), which is a transformer-based language model trained with two objectives: masked token prediction and next-sentence prediction. Although BERT has been applied to LS (Zhou et al., 2019; Qiang et al., 2019), these studies have focused on a single word as a simplification unit. The applicability of BERT in simplifying longer spans of text remains to be investigated.

3 Proposed Method

3.1 Overall Simplification Process

Following the general LS process (Shardlow, 2014), our proposed simplification process comprises the following four steps: detection of complex predicates, substitution generation, substitution validation, and substitution ranking. The overall process is shown in Figure 2. Three types of functional expressions ("*de aru*," "*to naru*" and "*ni naru*") in the input sentences, which in many cases unnecessarily complicate the sentence structures, are to be normalized in advance.

[2] https://simple.wikipedia.org/

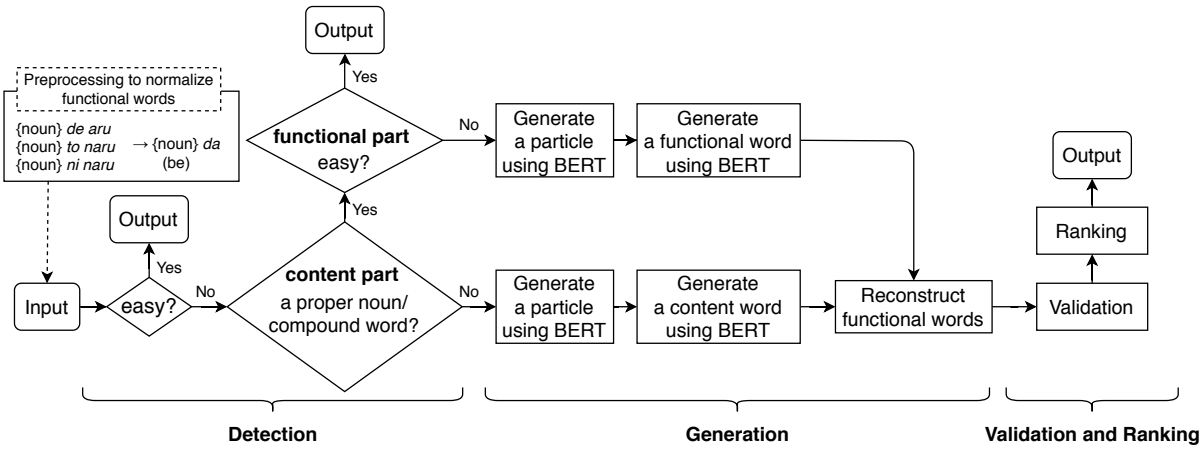

Figure 2: Overall process of simplification of sentence-ending predicates.

Detection Step The span of the sentence-ending predicate is first identified using the Japanese predicate analyzer Panzer (Sano et al., 2020). As shown in Figure 1, it basically comprises a particle, content part (verbs, nouns, adjectives and adverbs) and functional part (a sequence of functional words).[3] The difficulty of the predicate is then assessed based on the following two criteria, which correspond approximately to the level of Japanese around 10-years-old:

- Verbs, nouns, adjectives and adverbs in the content part are included in the vocabulary up to Level 2 of the Japanese Language Proficiency Test (JLPT).[4]
- Linguistic patterns (a sequence of functional words) in the functional part are included in the grammar items up to Level 3 of the JLPT.

Predicates that do not satisfy the criteria above are detected as difficult. At this stage, if the content part includes a proper noun or a compound word, it is excluded from the substitution generation step because it often represents an important concept that should not/cannot be simplified. In such cases, only the functional part is targeted for substitution if it is assessed as difficult. We used JUMAN++[5] to identify proper nouns and compound words.

Generation Step For the detected difficult phrase, simpler candidate phrases are generated using the proposed BERT-based method, the core mechanisms of which will be described in Section 3.2. Whereas two different inputs exist depending on the results of the detection step, this process generates two words: if the content part includes a proper noun or a compound word, it generates a particle and a functional word; otherwise, a particle and a content word.

The candidates generated by BERT are normalized and restored as complete sentences with reference to the original phrase using the Japanese text generation library HaoriBricks3 (Sato, 2020). In this study, the reconstructed elements are tense (*ta*-form), aspects (*teiru*-form), passive voice (*reru/rareru*-form), honorifics (*desu/masu*-form), and negative form (*nai*-form).

Validation and Ranking Steps Finally, the generated candidates are validated and ranked in terms of simplicity, fluency, and adequacy. Candidates that do not satisfy the aforementioned difficulty criteria are excluded, which ensures that all the final candidates are simpler than the original predicate. Subsequently, they are ranked using three features, BERT likelihood, cosine similarity, and language model perplexity, which we will describe in Section 3.3.

3.2 BERT-Based Predicate Generation

The input representation of the BERT is constructed by summing the token, segment, and position embeddings. To obtain a word whose meaning is similar to that of the original predicate *as a whole*, we used an average token embedding of the predicate as shown in Figure 3. We then used BERT to sequentially generate two words, (1) a particle and (2) a content or functional word.

[3] Linguistically, Japanese particles attach to the preceding words, but, based on our observation, it is useful to jointly include them in sentence-ending predicates.

[4] https://www.jlpt.jp/

[5] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN++

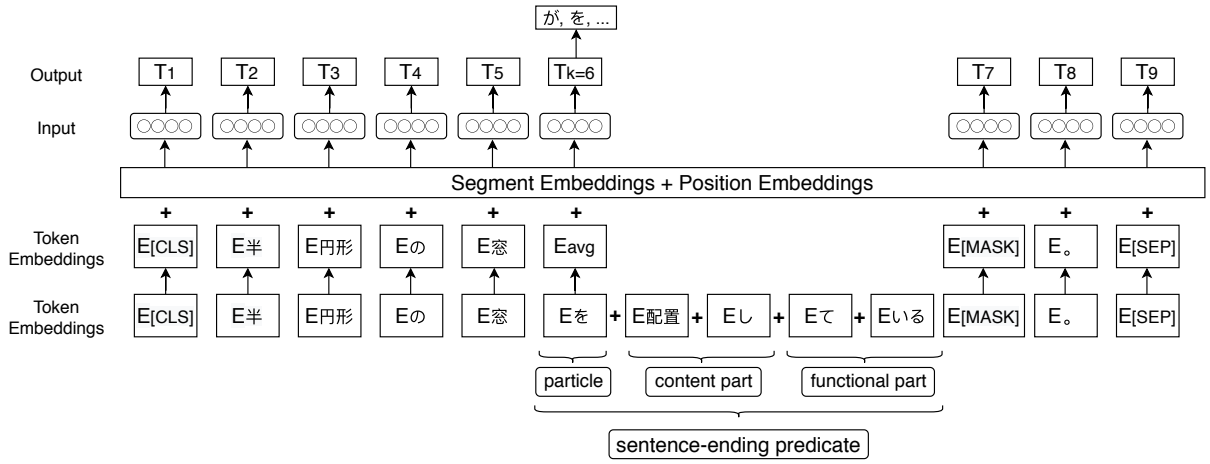

Figure 3: Generation of a particle by BERT with average token embedding. [CLS] and [SEP] are two special tokens in BERT; [CLS] is added in front of input tokens and [SEP] is a special separator token.

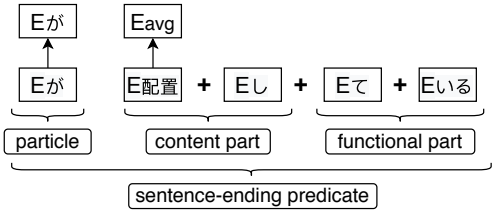

Figure 4: Generation of a content word.

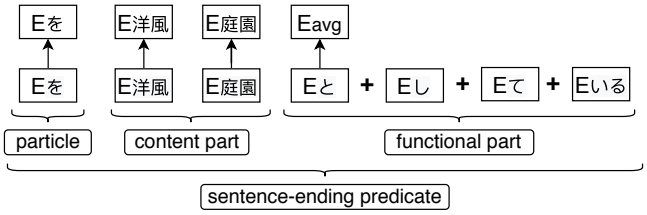

Figure 5: Generation of a functional word.

Generation of Particles As shown in Figure 3, average token embedding is used to represent the original predicates, including the particle. Let $\boldsymbol{w} = (w_1, ..., w_k, ..., w_L)$ be the tokens of a sentence of length L, where w_k is a particle at the head of the target phrase. We used token embeddings of sequence w_k to w_L to obtain the average token embedding of the predicate, which is calculated as follows:

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

We added the special token [MASK] and a Japanese period (。) at the end of the sentence. The [MASK] token is important because it functions as a pseudo content word, which can increase the probability of generating a particle. We removed tokens (w_{k+1},..., w_L) from $\boldsymbol{w}$ and added tokens ($w_{[MASK]}$, $w_{。}$); therefore, $\boldsymbol{w}' = (w_1, ..., w_k, w_{[MASK]}, w_{。})$ became a new sequence for the BERT input. Subsequently, we replaced token embedding E_k with $E_{avg[particle]}$.

The particles selected significantly affect the generation of a subsequent word. To obtain a diverse candidate generation, it is crucial to obtain multiple particles at this stage. Our proposed process generates 10 outputs in the descending order of their BERT likelihoods and retain up to two most likely particles.

Generation of Content/Functional Words Figures 4 and 5 depict the process of generating a content or a functional word subsequent to the particle, respectively.[6] The input tokens can now be re-expressed as $\boldsymbol{w} = (w_1, ..., w'_k, ..., w_L)$, where w'_k is the generated particle. The average token embeddings for a content word and a functional word are calculated as follows:

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

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

To obtain a wide range of outputs, we introduced an embedding dropout mechanism (Zhou et al., 2019), which partially causes the target input embedding to be zero. This can loosen the semantic relation between the original phrase and a word to be predicted, which may expand the diversity of the outputs.

[6] As described in Section 3.1, if the content part includes a compound word or a proper noun, then only the functional part is substituted.

In this study, for each generated particle, our process generates five outputs in descending order of their BERT likelihoods. It is noteworthy that, in rare cases, the BERT might produce undesirable outputs, such as ungrammatical particles; however, we do not filter them out because it is more important to obtain a wide range of candidates. We can manage such cases in the candidate ranking step. Finally, the original predicate is replaced with the generated candidates. As described in Section 3.1, the generated sentences are normalized and restored as complete sentences, and the difficult sentences are excluded.

3.3 Candidates Ranking

At this stage, 10 candidates are prepared at the most. To rank the candidates in terms of adequacy (meaning preservation) and fluency, we adopted the average rank of the following three features. The candidates with the same rank are reranked in the descending order of BERT likelihood.

Bert Likelihood The likelihood score was produced by BERT when each candidate was generated. We assume this is the most basic feature to measure the "goodness" of the candidates, as it may indicate how well the generated candidate fits in the context, how relevant it is to the original word (adequacy), and how natural it is as a Japanese expression (fluency). The averaged likelihoods of a generated particle and a content/functional word were used.

Cosine Similarity The cosine similarity of the embedding vectors between the original predicate and the generated word were used to calculate the semantic similarity (adequacy). If a content word is generated, the similarity between the content word and the original content part is measured. If only a functional part is generated, the similarity between the functional part and the original functional part is measured. For any parts having more than one word, the average of the embedding is used. To obtain the embedding vectors, we used existing Japanese pre-trained word vectors that were trained on Common Crawl and Wikipedia using fastText.[7]

Language Model Perplexity The scores of the language model were used to measure the fluency in Japanese. We built a transformer-based language model trained from a whole text from Japanese Wikipedia[8] using fairseq toolkit (Ott et al., 2019). For Japanese tokenization, we used MeCab[9] and sentencepiece.[10] The lower perplexity indicated a better result, that is, a higher rank.

4 Experiment

4.1 Dataset

We used the following three types of Japanese descriptive text for evaluation: Wikipedia texts,[11] news articles,[12] and expository text on cultural assets that were extracted from leaflets and websites of Japanese cultural assets.[13] For each domain, we randomly extracted 500 sentences. In the detection step, 704 of 1500 predicates were first identified as difficult. Subsequently, 160 phrases were excluded for simplification as they contained a proper noun or compound word with an easy functional part. Hence, as evaluation data, we obtained 544 sentences with difficult predicates to be simplified.

4.2 Methods

BERT-Based Method (Proposed) We tested three BERT-based methods: **BERT[Avg+dp]**, which uses both the average token embedding and dropout; **BERT[Avg]**, which uses only the average token embedding; and **BERT[MASK]**, which uses a special [MASK] token instead of the average token embedding. We used the Japanese BERT model,[14] whose tokenizer is MeCab+WordPiece, and set the dropout ratio to 0.3 following Zhou et al. (2019).

Thesaurus-Based Method (Baseline) As a baseline, we also implemented a thesaurus-based LS method that substitutes difficult content words into simpler ones. This method was integrated into our overall process in place of BERT-based generation modules in Figure 2. To obtain Japanese synonyms, we used three popular thesauri, i.e., Japanese WordNet Synonyms Database,[15] *Bunrui*

[7] https://fasttext.cc/docs/en/crawl-vectors.html

[8] The data were obtained from Wikimedia (https://dumps.wikimedia.org/).

[9] http://taku910.github.io/mecab/

[10] https://github.com/google/sentencepiece

[11] https://dumps.wikimedia.org/

[12] https://www3.nhk.or.jp/news/

[13] We treated highly technical terms contained in the expository text on cultural assets as proper nouns using the existing terminology of Japanese cultural assets (Kyoto bunkazai hogo kikin, 1989).

[14] https://github.com/cl-tohoku/BERT-japanese: BERT-base_mecabipadic-bpe-32k_whole-word-mask

[15] http://compling.hss.ntu.edu.sg/wnja/

	Fluency	Adequacy (meaning preservation)
1	Grammatically correct	The core meaning of the sentence is correctly conveyed.
2	Slightly awkward	The core meaning of the sentence may be conveyed depending on the context of usage.
3	Grammatically incorrect	The core meaning of the sentence is not conveyed.

Table 1: Human evaluation metrics.

Sets	# of unique sentences	Fluency	Adequacy
A	3480	0.733	0.707
B	1739	0.831	0.846

Table 2: Weighted Cohen's Kappa coefficients.

Methods	Total	Good	Acceptable
Thesaurus-based	641	111/641	183/641
BERT[MASK]	3158	439/3158	844/3158
BERT[Avg]	3031	451/3031	817/3031
BERT[Avg+dp]	3059	459/3059	853/3059

Table 3: Number of generated candidates.

Methods	Total	Good	Acceptable
Thesaurus-based	294 (0.540)	98 (0.180)	135 (0.248)
BERT[MASK]	540 (0.993)	221 (0.406)	362 (0.665)
BERT[Avg]	539 (0.991)	239 (0.439)	368 (0.676)
BERT[Avg+dp]	540 (0.993)	247 (0.454)	380 (0.699)

Table 4: Number of target sentences that have any good or acceptable candidate, with the ratio to the 544 target sentences given in parentheses.

Goi Hyo (Word List by Semantic Principles),[16] and SNOW D2 (Lexical Paraphrase Lexicon of Japanese Content Words).[17] For candidate ranking, we used the two features, cosine similarity and language model perplexity (excluding BERT likelihood).

4.3 Human Evaluation

Because a large-scale human gold standard for the simplification of Japanese predicates is unavailable, we adopted human evaluation. We recruited four native speakers of Japanese as evaluators. We divided the output sentences generated via the four methods into two sets (A and B) and respectively assigned two evaluators. Each sentence was evaluated on the basis of two metrics, i.e., fluency and adequacy, as shown in Table 1.

Table 2 shows the inter-rater agreement scores, i.e., the weighted Cohen's Kappa coefficients (Cohen, 1968), between two evaluators for each evaluation set and metric. All scores were above 0.7, indicating a fairly reliable agreement (Artstein and Poesio, 2008).

5 Results and Analyses

5.1 Results

We regarded candidates whose fluency and adequacy were rated as 1 by the two evaluators as *good* candidates, and those whose fluency and adequacy were rated as 1 by at least one evaluator as *acceptable* candidates.[18]

Table 3 shows the statistics of the generated candidates. The total numbers of candidates generated by the BERT-based methods were much higher than that by the thesaurus-based method. The results also show that the BERT-based methods generated more good and acceptable candidates than the thesaurus-based method.

Table 4 shows the statistics of target sentences that obtained any candidate. It is noteworthy that for most of the target sentences, the BERT-based methods generated at least one candidate. Meanwhile, the thesaurus-based method, which is constrained by the amount of language resources, could not generate any candidate for 46% of the sentences. Moreover, the total numbers of sentences having acceptable candidates via the BERT-based methods were nearly three times that obtained via the thesaurus-based method. These results strongly demonstrate the powerful generation ability of BERT compared with the conventional method. This suggests that the basic architecture of the masked language model is crucial.

Acceptable@N shown in Table 5 is the rate at which any acceptable candidate is included in the top-N candidates of the ranked list. Acceptable@1 is an important indicator for developing a fully automatic simplification system because the system must select only one candidate, whereas Acceptable@5 provides useful insight into the application to the simplification support system that provides a list of candidates for human writers. The left part of Table 5 shows the results when all the ranking features were used. The results of Acceptable@1 show that even though the BERT-based methods

[16] https://github.com/masayu-a/WLSP
[17] http://www.jnlp.org/SNOW/D2
[18] *Acceptable* candidates include *good* candidates.

Methods	All ranking features		-w/o LM perplexity		-w/o cosine similarity		-w/o BERT likelihood	
	@1	@5	@1	@5	@1	@5	@1	@5
Thesaurus-based	**112 (0.206)**	135 (0.248)	97 (0.178)	135 (0.248)	98 (0.18)	135 (0.248)	-	-
BERT[MASK]	**254 (0.467)**	355 (0.653)	251 (0.461)	355 (0.653)	246 (0.452)	357 (0.656)	247 (0.454)	352 (0.647)
BERT[Avg]	**255 (0.469)**	365 (0.671)	246 (0.452)	362 (0.665)	241 (0.443)	362 (0.665)	246 (0.452)	362 (0.665)
BERT[Avg+dp]	263 (0.483)	373 (0.686)	**267 (0.491)**	374 (0.688)	258 (0.474)	371 (0.682)	256 (0.471)	373 (0.686)

Table 5: Results of Acceptable@N (@1 and @5) and effectiveness of each ranking feature, with the ratio to the 544 target sentences given in parentheses.

Input	正面に下屋**が設けられています**。 *Shōmen ni geya* ***ga mōke rarete imasu***. (The small roof **is set** at the front.)				
Thesaurus-based	[No candidate was generated]				
BERT[MASK]	**<u>があります</u>** ***<u>ga ari masu</u>*** **(there is)**	**<u>を備えています</u>** ***<u>wo sonaete imasu</u>*** **(equip)**	?を持っています *wo motte imasu* (have)	?を置かれています *wo okareteimasu* (be placed)	**<u>が付いています</u>** ***<u>ga tsuite imasu</u>*** **(be attached)**
BERT[Avg]	**<u>があります</u>** ***<u>ga ari masu</u>*** **(there is)**	**<u>を備えています</u>** ***<u>wo sonaete imasu</u>*** **(equip)**	?を持っています *wo motte imasu* (have)	?をもっています *wo motte imasu* (have)	**<u>が付いています</u>** ***<u>ga tsuite imasu</u>*** **(be attached)**
BERT[Avg+dp]	**<u>があります</u>** ***<u>ga ari masu</u>*** **(there is)**	**<u>を備えています</u>** ***<u>wo sonaete imasu</u>*** **(equip)**	?を持っています *wo motte imasu* (have)	?が入っています *ga haitte imasu* (be placed)	**<u>が付いています</u>** ***<u>ga tsuite imasu</u>*** **(be attached)**

Table 6: Example of the advantage of BERT-based methods over the thesaurus-based method. Top-5 outputs are shown (boldface: acceptable candidates; underline: good candidates; "?" preceding the candidates: awkward or ungrammatical candidates).

delivered significantly higher performance than the thesaurus-based method, the scores were in the range of 0.4–0.5, indicating opportunity for improvement. Meanwhile, BERT[Avg+dp] achieved 0.686 for Acceptable@5, which implies that our proposed method can be useful for providing human writers with candidates for selection.

5.2 Analyses

5.2.1 Detailed Comparison of Methods

Table 6 shows an example of a Japanese predicate and part of the generated candidates. Whereas the thesaurus-based method could not generate even one candidate, all the BERT-based methods can generate some candidates and good candidates "*ga arimasu* (there is)" were ranked top. Even though the generated phrase "*ga arimasu* (there is)" and the original phrase "*ga mōke rareteimasu* (be arranged)" are not synonyms *per se*, they become synonymous when the entire sentence is considered. Importantly, all of the BERT-based methods produced the similar candidate lists. This suggests that the masked language model functions well and simulates human intuitions for predicting omitted predicates by referring only to the preceding context, as mentioned in Section 1. Meanwhile, the thesaurus-based method, which does not take into account the context, could not generate such candidates.

To examine the effectiveness of the usage of the average token embedding and dropout, with a focus on the sentences that obtained at least one acceptable candidate in Table 4, we applied McNemar's test for all combinations of the results for the BERT-based methods. The p-values for BERT[Avg]–BERT[Avg+dp] and BERT[MASK]–BERT[Avg+dp] were 0.0357 and 0.0198 respectively, while that for BERT[MASK]–BERT[Avg] was 0.5504. This implies that the combination of the average token embedding and dropout is particularly important for this simplification task.

Table 7 shows an example of the effectiveness of average token embedding; the target predicate "*ga zōkasuru* (increase)" is difficult to determine only from the context, and BERT[MASK], which does not use the information of the original phrase, could not generate any good candidate. In such cases, the average token embedding that encodes the information of the original can improve the performance. Furthermore, whereas the thesaurus-based method generated one good candidate, BERT[Avg] generated four good candidates in the top-5 list.

Input	ハムストリングスなどの筋量**が増加する** *Hamusutoringusu nadono kinryō **ga zōkasuru*** (The amount of muscle such as hamstrings **increases**.)				
Thesaurus-based	**が増す** ***ga masu*** **(increase)**				
BERT[MASK]	を測定する *wo sokuteisuru* (measure)	を示す *wo shimesu* (indicate)	を測る *wo hakaru* (measure)	を表す *wo arawasu* (show)	を用いる *wo mochīru* (use)
BERT[Avg]	が多い *ga ōi* (many)	**が増える** ***ga fueru*** **(increase)**	**を増やす** ***wo fuyasu*** **(increase)**	**を高める** ***wo takameru*** **(increase)**	**が増す** ***ga masu*** **(increase)**

Table 7: Example of the effectiveness of average token embedding. Top-5 outputs are shown (boldface: acceptable candidates; underline: good candidates).

Input	調査開始以来最も多くなる**見通しです**。 *Chōsa kaishi irai mottomo ōku naru **mitōshi desu**.* (This **is predicted to** be the highest number since the survey began.)			
Bert[MASK]	数字です *sūji desu* (is the number)	記録です *kiroku desu* (is the record)		
Bert[Avg]	**予定です** ***yotei desu*** **(is planned to)**	?です *desu* (is)	?ます *masu* (is)	
Bert[Avg+dropout]	**予定です** ***yotei desu*** **(is planned to)**	水準です *suijun desu* (is the number)	**予測です** ***yosoku desu*** **(is expected to)**	

Table 8: Example of the effectiveness of average token embedding and dropout (boldface: acceptable candidates; underline: good candidates; "?" preceding the candidates: awkward or ungrammatical candidates).

Table 8 shows an example of the effectiveness of average token embedding and dropout. BERT[MASK], which does not use the information of the original predicate, could not generate any acceptable candidate; however, the other two models, which encode the information of the original predicate by using the average token embedding, generated acceptable candidates. Moreover, only BERT[Avg+dp] successfully generated a good candidate. BERT[Avg+dp] partially encodes target tokens and can generate wider candidates similar to those part of the original expressions.

5.2.2 Error Analysis

To understand the limitations of our methods and devise ways to improve their performances, we analyzed the cases in which all of the BERT-based methods could not generate any acceptable candidates. We manually classified all such cases, totaling 140 cases, in a bottom-up manner. Table 9 presents the constructed error categories.

Two error types were relevant to the detection step (see also Figure 2): incorrect identification of phrasal span and incorrect identification of proper nouns/compound words. The former type was often caused by idiomatic expressions. As shown in Table 9, the predicate "*isai wo hanatte imasu* (stands out conspicuously)" is an idiomatic expression that should be treated as one unit. However, our detection process separated this expression and detected only "*wo hanatte imasu*" as a predicate. Because it is impossible to substitute only one part of an idiom, idiomatic dictionaries are required to achieve a more accurate identification of predicates. The latter type of error is attributable to errors in upstream natural language processing tools and dictionaries. Hence, it is important to incorporate a mechanism for their automatic identification to mitigate the insufficiency in language pre-processing.

The generation step contains three error types. The first error type includes cases in which our method could *theoretically* generate some candidates[19] but failed to do so. To further increase the generation capability, we must consider the use of other models or the construction of a large phrase-level paraphrase dictionary. The next error type stems mainly from the deficiency in the reconstruction process of functional parts. Although our method can reconstruct the four functional words of JLPT level 4 (the easiest level) and the one functional word of JLPT level 3, we may need to expand

[19]To ascertain the *theoretical* possibility, we attempted to create any candidates using human linguistic introspection.

Process	Error category	#	Example
Detection	Incorrect identification of phrasal span	2	戸口を設けるなど異彩を放っています。 *Toguchi wo mōkeru nado **isai wo hanatte imasu**.* (It **stands out conspicuously** as the door is installed.)
	Incorrect identification of proper nouns/compound words	7	安祥寺から移して本尊としました。 *Anshōji kara utsushite **honzon to shimashita**.* (It was moved from the Anshoji temple and **became the principal deity**.)
Generation	No acceptable candidate	99	風船を空に向かって放ちました。 ***Fūsen wo sora ni mukatte hanachi mashita**.* *(They **released** the balloons up into the sky.)*
	Failure to reconstruct functional part	2	１人30グラムまで所持することができます。 *1-Ri 30-guramu **made shoji suru koto ga dekimasu*** (Each person **can possess up to** 30 grams.)
	Architectural incapability	30	連合国の無線を傍受していた。 *Rengōkoku no musen **wo bōju shite ita**.* (They **were intercepting** the Allies' radio messages.)

Table 9: Error category for cases in which no acceptable candidate was generated.

them. However, an excessive reconstruction of the functional part may induce textual difficulty, which we intend to avoid. The last error type is architectural incapability. The predicate "*wo bōju shite ita* (were intercepting)" in the example sentence can be rewritten as "*wo kakurete kiite ita* (were secretly listening to)"; however, they comprise two content words. Because our method can generate only one content word, it must be extended to address this problem. For example, a method can be developed that does not place a period at the end of the sentence and continues to generate words until BERT outputs a period. However, more generated words result in more combinations; hence, the control and ranking of the candidates become more difficult.

6 Conclusion

In this paper, we proposed a BERT-based method for generating simplified substitutions for Japanese sentence-ending predicates in descriptive text. The principal feature of our method is that it can generate candidate predicates that conform to the target context and retain the core meaning of the original one, even though the candidates themselves may not always be exactly synonymous to the original one. Furthermore, our method does not rely on conventional language resources such as thesauri but utilizes a large pre-trained language model and can generate an output for almost any input.

The evaluation results showed that the proposed method generated simple, fluent candidates that conveyed the original meaning for 54% of difficult predicates and substantially outperformed the conventional thesaurus-based method. Furthermore, we investigated the effectiveness of using average token embedding and dropout in the BERT model and systematically analyzed the critical errors.

In future studies, we plan to extend our architecture to manage idiomatic expressions and produce multiple content words to improve expressiveness. We will then develop a simplification support system integrating our proposed method that can return a list of candidates for human writers. We will also aim to examine our method for other text types, such as narrative and argumentative texts (Robert and Wolfgang, 1981). In this regard, important information is encoded in predicates, which renders the task more challenging.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 19K20628 and by the Research Grant Program of KDDI Foundation, Japan. The human evaluation work was supported by BAOBAB Inc.

References

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213–220.

Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Proceedings of the 2010 SIGIR Workshop on Accessible Search Systems*, pages 19–26, New York.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Goran Glavaš and Sanja Štajner. 2015. 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*, pages 63–68, Beijing, China.

Akihiro Katsuta and Kazuhide Yamamoto. 2018. Crowdsourced corpus of sentence simplification with core vocabulary. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, pages 461–466, Miyazaki, Japan.

Daisuke Kawahara, Sadao Kurohashi, and Kôiti Hasida. 2002. Construction of a Japanese relevance-tagged corpus. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 2008–2013, Las Palmas, Canary Islands.

Kyoto bunkazai hogo kikin, editor. 1989. *Bunkazai Yougo Jiten [Terminology of Cultural Assets]*. Tankosha, Kyoto. (in Japanese).

Daiki Nishihara and Tomoyuki Kajiwara. 2020. Word complexity estimation for Japanese lexical simplification. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3114–3120.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 85–91, Vancouver, Canada.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota.

Gustavo H Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the 30th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, pages 3761–3767, Phoenix, Arizona, USA.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple PPDB: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 143–148, Berlin, Germany.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. 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*, pages 425–430, Beijing, China.

Jipeng Qiang, Yun Li, Yi Zhu, and Yunhao Yuan. 2019. A simple bert-based approach for lexical simplification. *arXiv preprint arXiv:1907.06226*.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey.

Beaugrande Robert, de and Dressler Wolfgang. 1981. *Introduction to Text Linguistics*. Routledge, New York.

Thomas S. Rebecca and Anderson Sven. 2012. WordNet-based lexical simplification of a document. In *Proceedings of the 11th Conference on Natural Language Processing (KONVENS)*, pages 80–88.

Masahiro Sano, Satoshi Sato, and Rei Miyata. 2020. Detection of functional expressions in Japanese sentence-ending predicative phrases and its application to estimation of rhetorical relation between sentences. In *Proceedings of the 26th Annual Meeting of the Association for Natural Language Processing*, pages 1483–1486, Online. (in Japanese).

Satoshi Sato. 2020. HaoriBricks3: A domain-specific language for Japanese sentence composition. *Journal of Natural Language Processing*, 27(2):411–444. (in Japanese).

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications, Special Issue on Natural Language Processing 2014*, 4(1):581–701.

Sanja Štajner and Maja Popovic. 2016. Can text simplification help machine translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 230–242, Riga, Latvia.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 1015–1024, Jeju Island, Korea.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1353–1361, Beijing, China.