Contextualized *context2vec*

Kazuki Ashihara
Osaka University
ashihara.kazuki@ist.osaka-u.ac.jp

Tomoyuki Kajiwara
Osaka University
kajiwara@ids.osaka-u.ac.jp

Yuki Arase
Osaka University
arase@ist.osaka-u.ac.jp

Satoru Uchida
Kyushu University
uchida@flc.kyushu-u.ac.jp

Abstract

Lexical substitution ranks substitution candidates from the viewpoint of paraphrasability for a target word in a given sentence. There are two major approaches for lexical substitution: (1) generating contextualized word embeddings by assigning multiple embeddings to one word and (2) generating context embeddings using the sentence. Herein we propose a method that combines these two approaches to contextualize word embeddings for lexical substitution. Experiments demonstrate that our method outperforms the current state-of-the-art method. We also create CEFR-LP, a new evaluation dataset for the lexical substitution task. It has a wider coverage of substitution candidates than previous datasets and assigns English proficiency levels to all target words and substitution candidates.

1 Introduction

Lexical substitution (McCarthy and Navigli, 2007) is the finest-level paraphrase problem. It determines if a word in a sentence can be replaced by other words while preserving the same meaning. It is important not only as a fundamental paraphrase problem but also as a practical application for language learning support such as lexical simplification (Paetzold and Specia, 2017) and acquisition (McCarthy, 2002). Table 1 shows an example of the lexical substitution task with a sentence,[1] the target word to replace, and words of substitution candidates. The numbers in parentheses represent the paraphrasability of each candidate, where a larger value means the corresponding word is more appropriate to substitute the target word. The lexical substitution task ranks these candidates according to assigning weights. The key technology to solve lexical substitution tasks is to precisely capture word senses in a context.

context	... explain the basic concept and purpose and get it **going** with minimal briefing .
target	go
candidate	start (4), proceed (1), move (1) ...

Table 1: Example of the lexical substitution tasks

There are mainly two approaches for lexical substitution: (1) generating contextualized word embeddings by assigning multiple embeddings to one word and (2) generating context embeddings using the sentence. The former realizes static embeddings as it pre-computes word embeddings. One example of the first approach is *DMSE* (Dependency-based Multi-Sense Embedding), which was proposed by Ashihara et al. (2018) to contextualize word embeddings using words with dependency relations as a clue to distinguish senses. As an example of the second approach, *context2vec* (Melamud et al., 2016) generates a context embedding by inputting the sentence into bidirectional recurrent neural networks. It combines context embedding and a simple word embedding to generate a dynamic embedding. These two methods are current state-of-the-arts among methods of each approach.

We focus on the fact that these two methods have a complementary nature. *DMSE* considers only a single word as context, while *context2vec* uses a simple word embedding. Herein we combine *DMSE* and *context2vec* to take advantages of both contextualized word embeddings and context embeddings. Specifically, we apply a contextualized word embedding generated by *DMSE* to replace the word embedding used in *context2vec*.

[1] In this paper, the terms *context* and *sentence* are used interchangeably wherever the context for the target refers to the sentence.

Proceedings of the 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-generated Text, pages 397–406
Hong Kong, Nov 4, 2019. ©2019 Association for Computational Linguistics

In addition, we create a new evaluation dataset for lexical substitution, named CEFR-LP. It is an extension of CEFR-LS (Uchida et al., 2018) and is created for lexical simplification to support substitution tasks. The benefits of CEFR-LP are that it expands the coverage of substitution candidates and provides English proficiency levels. These features are unavailable in previous evaluation datasets such as LS-SE (McCarthy and Navigli, 2007) and LS-CIC (Kremer et al., 2014).

The evaluation results on CEFR-LP, LS-SE, and LS-CIC confirm that our method effectively strengthens *DMSE* and *context2vec*. Additionally, our proposed method outperforms the current state-of-the-art methods. The contributions of this paper are twofold:

- A method that takes advantages of contextualized word embedding and dynamic embedding generation from contexts is proposed. This method achieves a state-of-the-art performance on lexical substitution tasks.
- Creation and release[2] of CEFR-LP, which is a new evaluation dataset for lexical substitution with an expanded coverage of substitution candidates and English proficiency levels.

2 Related Work

There are two major approaches to lexical substitution. One approach generates contextualized word embeddings by assigning multiple embeddings to one word. Paetzold and Specia (2016) generated word embeddings per part-of-speech of the same word assuming that words with the same surface have different senses for different part-of-speech. Fadaee et al. (2017) also generated multiple word embeddings per topic represented in a sentence. For example, the word `soft` may have embeddings for topics of food when used like `soft cheese` and that for music when used like `soft voice`. To adequately distinguish these word senses, both methods assign embeddings that are too coarse. For example, the phrases `soft cheese` and `soft drink` both use `soft` as an adjective and are related to the food topic. The former has the sense of `tender` while the latter represents the sense of `non-alcoholic`. To solve this problem, *DMSE* generates finer-grained word embeddings because it generates embeddings for words with dependency relations based on the CBOW algorithm of word2vec (Mikolov et al., 2013). It concatenates words with dependent relations within a specific window, which is a hyperparameter in CBOW. Hence considered context in *DMSE* is bounded by the window size. *DMSE* achieves the highest performance for lexical substitution tasks among the methods categorized into the first approach.

The other approach dynamically generates contextualized embeddings considering a sentence. *Context2vec* generates a context embedding using bidirectional long short-term memory (biLSTM) networks (Schuster and Paliwal, 1997). Then it combines the context embedding with a simple word embedding. *Context2vec* is the current state-of-the-art method for representative lexical substitution tasks. Its advantage is that it can consider the entire sentence as the context, while *DMSE* is bounded by a window size. However, *DMSE* can use contextualized word embeddings, whereas *context2vec* just uses a simple word embedding for each word. The complementary nature of these two methods inspired us to combine them. More recently, *ELMo* (Peters et al., 2018) showed a language modeling using biLSTM networks produces contextualized word embeddings, which are effective for various NLP tasks such as named entity recognition. *Context2vec* differs from *ELMo* when explicitly considering word embeddings of substitution targets. Our experiments empirically confirm that *context2vec* outperforms *ELMo* in Section 6.

3 Proposed Method

We combine *DMSE* and *context2vec* to take advantage of both fine-grained contextualized word embeddings and context embeddings.

3.1 Overview

DMSE is designed to train its word embeddings using CBOW, which we replaced with biLSTM networks in *context2vec*. *DMSE* contextualizes a word using words with dependency relations (both head and dependents) in a given sentence. Hereafter, words with dependency relations are referred to as *dependency-words*.[3]

There are numerous number of combinations

[2] `http://www-bigdata.ist.osaka-u.ac.jp/arase/pj/CEFR-LP.zip`

[3] These are called as *context-words* in Ashihara et al. (2018).

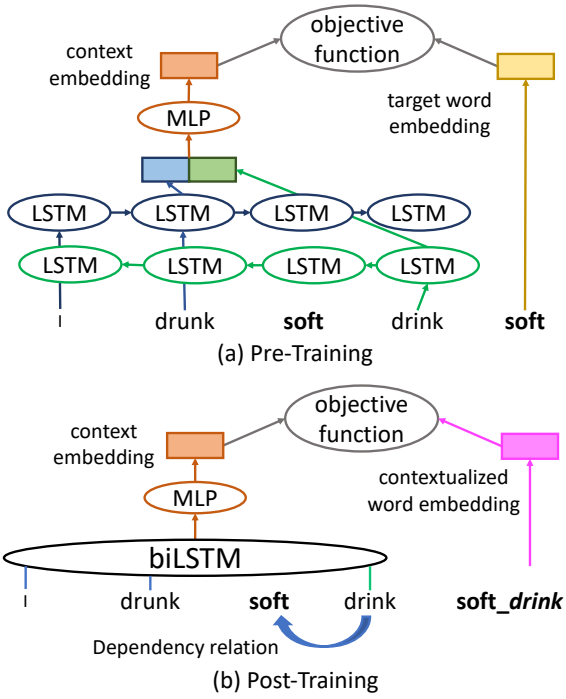

Figure 1: Design of the proposed method where `soft` is the target to generate embedding.

of words and *dependency-words*. Similar to Ashihara et al. (2018), we implement a two-stage training: pre-training and post-training for computational efficiency. In the pre-training, simple word embeddings (one embedding per word) and parameters of biLSTM networks are trained by *context2vec*. In the post-training, only contextualized word embeddings are trained starting from the pre-trained word embeddings.

3.2 Pre-Training

Figure 1 (a) overviews pre-training, which corresponds to the training of *context2vec*. Word embeddings and parameters of biLSTM networks are set.

First, the entire sentence is inputted into the biLSTM networks. At time step k, the forward network encodes words from the beginning to the k-th word. The backward network does the same except in the opposite direction. Therefore, the outputs of each LSTM network before and after a target word represent the preceding and following contexts surrounding the target word, respectively. These outputs are concatenated and inputted into a multi-layer perceptron to generate a unified context embedding for the target word. On the other hand, the target word is represented by a word embedding that has the same dimensions as the context embedding.

The objective function is the negative sampling proposed by Mikolov et al. (2013). A positive example is the target word and its context, whereas negative examples are random words. Note that word embeddings, forward LSTM network, and the backward LSTM network each have their own parameters.

3.3 Post-Training

Figure 1 (b) outlines post-training. Multiple word embeddings are generated for words with the same surface but with different *dependency-words* as contextualized word embeddings.

First, the sentence is parsed to obtain *dependency-words* of the target. For each *dependency-word* and target pair, its word embedding is trained. The process is simple. These words are concatenated with an under-bar (_) and treated as a single word, whose embedding is used as a contextualized word embedding of the target word. The contextualized word embeddings are trained in the same manner with pre-training.

The contextualized embeddings are initialized by assigning the pre-trained word embeddings in Section 3.2. The pre-trained word embeddings and biLSTM networks are fixed, and only the contextualized word embeddings are updated during post-training. This setting allows the contextualized embeddings to be trained in parallel.

3.4 Application to Lexical Substitution Task

This section describes how to tackle the lexical substitution task using both contextualized word embeddings and context embeddings obtained by the proposed method.

Ranking Method As shown in Table 1, lexical substitution ranks substitution candidates of the target word based on their paraphrasabilities under a given context. We use the same ranking method with *context2vec*, which assumes not only that a good substitution candidate is semantically similar to the target word but also is suitable for a given context. This assumption is commonly used in recent lexical substitution models (Melamud et al., 2015; Roller and Erk, 2016).

Here we have target word t and its *dependency-word* d. The contextualized word embedding of t is noted as $\boldsymbol{v}_t^d$ and the word embedding of a substitution candidate s contextualized by d is $\boldsymbol{v}_s^d$. Finally, the context embedding is denoted as $\boldsymbol{v}_c$. The

following scores are calculated for each substitution candidate and ranked them in descending order.

$$S(\boldsymbol{v}_s^d|\boldsymbol{v}_t^d, \boldsymbol{v}_c) = (\cos(\boldsymbol{v}_t^d, \boldsymbol{v}_s^d) + 1)(\cos(\boldsymbol{v}_s^d, \boldsymbol{v}_c) + 1). \quad (1)$$

Here, $\cos(\cdot, \cdot)$ calculates the cosine similarity between two vectors. If the word embedding does not exist in the vocabulary, the word embedding of $\langle \texttt{unk} \rangle$ is used.

***Dependency-word* Selection** When there are multiple *dependency-words* to contextualize a word embedding, the most appropriate one must be selected to characterize the sense of the target word in a given context. Ashihara et al. (2018) proposed the following *dependency-word* selection method for the *DMSE* model.

$$S_{\text{maxc}} : d = \arg\max_{d \in D} S(\boldsymbol{v}_s^d|\boldsymbol{v}_t^d, \boldsymbol{v}_c),$$

where D is a set of *dependency-words* of the target word in the context. If the contextualized word embedding $\boldsymbol{v}_s^d$ or $\boldsymbol{v}_t^d$ does not exist in the vocabulary, the corresponding simple word embeddings ($\boldsymbol{v}_s$ or $\boldsymbol{v}_t$) pre-trained for *context2vec* are used.

S_{maxc} uses the *dependency-word* that maximizes the paraphrasability score, but there is no guarantee that this *dependency-word* best characterizes the sense of the word in the given context. Therefore, we propose the following *dependency-word* selection methods based on the similarity between the target word or candidate words and the context.

$$S_{\text{tar}} : d = \arg\max_{d \in D} \cos(\boldsymbol{v}_t^d, \boldsymbol{v}_c),$$

$$S_{\text{can}} : d = \arg\max_{d \in D} \cos(\boldsymbol{v}_s^d, \boldsymbol{v}_c).$$

These methods should select more appropriate *dependency-word* using both contextualized word embeddings and context embeddings.

4 CEFR-LP: New Evaluation Dataset

In addition to proposing a method for lexical substitution, we created CEFR-LP, which mitigates limitations of previous evaluation datasets.

4.1 Principle of CEFR-LP

LS-SE (McCarthy and Navigli, 2007) and LS-CIC (Kremer et al., 2014) are the standard evaluation datasets for lexical substitution. However, they have limited annotation coverage because the annotators provide substitution candidates manually. Specifically, each annotator provides up to three substitution candidates for LS-SE and up to five substitution candidates for LS-CIC. These candidates are regarded as appropriate candidates for a target under a specific context. During an evaluation, these candidates are combined for the same targets with different contexts. This leads to two limitations. First, annotators may not derive all the appropriate candidates for the target. Second, some appropriate candidates for a target among the combined ones are regarded as inappropriate because they were missed by the anasnotators when annotating the target under the given context.

To mitigate these limitations, CEFR-LS (Uchida et al., 2018) was constructed to improve the coverage. However, the target is lexical simplification rather than substitution. Herein we extend CEFR-LS and build a new evaluation dataset called CEFR-LP for lexical substitution tasks that:

1. Define the substitution candidates
2. Determine the paraphrasability label
3. Evaluate the number of annotators

The first extension adapts to lexical substitution. CEFR-LS only includes substitutions from complex words to simpler ones because it is specifically intended for simplification. On the other hand, CEFR-LP includes not only complex to simple substitutions but also simple to complex substitutions and substitutions between equivalent complexities. The substitution candidates are a synonym set of target words extracted from a dictionary.[4] The second extension generates fine-grained judgments for paraphrasability. CEFR-LS is annotated with binary labels, while CEFR-LP is annotated with continuous values representing paraphrasability. This extension allows automatic evaluation via the Generalized Average Precision (GAP) score (Kishida, 2005; Thater et al., 2009), which is common in recent lexical substitution studies. The last extension reduces potential annotation biases. While CEFR-LS was annotated by one expert, CEFR-LP employs more than five annotators per target to reduce bias due to annotator subjectivity. Following CEFR-LS,

[4] http://www.thesaurus.com/

context	..., and to create elixirs to cure disease and extend life . From alchemy came the historical **progressions** that led to modern chemistry : the isolation of drugs from natural sources , metallurgy , and the dye industry . Today , chemistry continues to deepen our understanding ...
target	progressions [C1]
candidate	block [B1] (0), development [B1] (6), advancement [B2] (8), break [A2] (1), ...
context	... Competition would ensure that prices remained low and faulty goods disappeared from the market . In this way , businesses would reap profits , consumers would have their needs satisfied , and society as a whole would **prosper** . Smith discussed these ideas, ...
target	prosper [B2]
candidate	thrive [C1] (8), blossom [B2] (6), yield [B2] (1), bear [A2] (0), flourish [C2] (8), ...
context	... That is , a member of the population may be chosen only once . Most samples are taken from **large** populations and the sample tends to be small in comparison to the population . Since this is the case , sampling without replacement is approximately ...
target	large [A1]
candidate	substantial [B1] (8), giant [B1] (6), extravagant [C2] (0), wide [A2] (1), ...

Table 2: Examples of CEFR-LP. "Context" shows context sentences where the target word is presented in **bold**. "Target" shows the target and "candidate" lists the substitution candidates. Square brackets indicate CEFR levels of targets and candidates. Round brackets indicate the weights of candidates.

CEFR-LP also provides CEFR (the Common European Framework of Reference for Languages) levels (A1 (lowest), A2, B1, B2, C1, and C2 (highest)) for the target and candidates as English proficiency levels.

4.2 Annotation

Following CEFR-LS, we use sentences extracted from textbooks publicly available at the OpenStax website[5] initiated by Rice University. We hired annotators on Amazon Mechanical Turk,[6] who (1) possessed a degree from an accredited university in the United States and (2) held the Mechanical Turk Masters qualification or a past acceptance rate above 98%.

Annotators were given a target word, its context, and a list of synonyms. They annotated each substitution candidate in the synonym list with paraphrasability labels ("sure", "maybe", and "not possible") considering the given context. As the context, a sentence on which the target word appeared as well as two more sentences before and after it were provided. To avoid overloading the annotator, target words with more than 30 synonyms were excluded.

Following CEFR-LS, we used the following annotation criteria:

Grammatical Reformation Stage When paraphrasing the target word into the substitution candidate, grammatical accuracy such as the part-of-speech and the connection to the preposition must be maintained. The morphology of the target word such as past tense and third person singular are automatically corrected.

Definition Stage The target word and the substitution candidate have the same meaning.

Context Stage The candidate should retain the nuance of the target word in a given context and not affect the meaning of the sentence.

If all of the above conditions were met, a label of "sure" is assigned. If either condition was not met, a label of "not possible" was assigned. If the judgment was difficult, a label of "maybe" was assigned.

Each annotation set was assigned to at least five annotators. To improve the reliability of annotation labels, we discarded the result from the annotator who had the lowest agreement with the others. Consequently, each set had four annotators and the average Fleiss' kappa was 0.33.

To use CEFR-LP for a lexical substitution task, the assigned labels were consolidated as a weight. For example, LS-SE and LS-CIC were set such that a weight to the number of annotators produced

[5] https://cnx.org/
[6] https://www.mturk.com/

	CEFR-LP	LS-SE	LS-CIC
number of target words	863	2,010	15,344
number of substitution candidates	14,259	34,600	601,257
average number of substitution candidates per target	16.5	17.2	39.2
average number of paraphrasable candidates per target	10.0	3.48	6.65

Table 3: Basic statistics in CEFR-LP compared to LS-SE and LS-CIC

CEFR level	target	candidate
all	863	14,259
A1	300	2,090
A2	190	2,856
B1	110	4,513
B2	186	3,201
C1	30	648
C2	47	951

Table 4: Distribution of CEFR levels in CEFR-LP

context embedding units	300
LSTM hidden/output units	600
MLP input units	1200
MLP hidden units	1200
sentential context units	600
target word units	600
number of negative samples	10
negative sampling rate	0.75
number of epochs	10

Table 5: *Context2vec* hyper-parameters that show the best performance in (Melamud et al., 2016).

a certain candidate. A "sure", "maybe", and "not possible" label were assigned values of 2, 1, and 0 points, respectively. These values were summed to give the weight of the candidate. Because each substitution candidate has four annotation labels, the weight ranged from 0 to 8. The larger the value, the higher the paraphrase possibility.

Table 2 shows examples sampled from CEFR-LP. "Context" gives sentences, including a target word. "Target" is the target word with its CEFR level in a square bracket, which is represented by a bold style in the context sentences. "Candidate" lists substitution candidates with their CEFR levels in square brackets and weights computed based on annotated labels in round brackets.

4.3 Analysis of CEFR-LP

Table 3 shows the basic statistics for CEFR-LP compared to those in LS-SE and LS-CIC. CEFR-LP provides 14,259 substitution candidates for 863 target words. The average number of paraphrasable candidates per word is 10.0, which is larger than 3.48 of LS-SE and 6.65 of LS-CIC. Here, a paraphrasable candidate means substitution candidates with a weight of 1 or more (*i.e.,* at least one annotator judged it can paraphrase the target in a given context). Compared to LS-SE and LS-CIC, CEFR-LP has an enhanced coverage of substitution candidates.

Table 4 shows the distribution of CEFR levels in CEFR-LP. Words at the C1 and C2 levels are naturally less frequent than others in general documents. The distribution reflects this tendency. We believe that these CEFR levels are useful when applying lexical substitution technologies to educational applications.

5 Evaluation Settings

This section describes the evaluation settings used to investigate the performance of our method on lexical substitution tasks.

5.1 Training of Our Method

To train contextualized word embeddings by using our method, we used 61.6M sentences[7] extracted from the main contents of English Wikipedia[8] articles. We lemmatized each word using the Stanford Parser (Manning et al., 2014) and replaced words less than or equal to ten frequency to ⟨unk⟩ tag to reduce the size of the vocabulary.

Pre-training used the same hyper-parameter settings of *context2vec* (Table 5). These settings achieved the best performance on lexical substitution tasks in Melamud et al. (2016).

[7]To speed-up the training of *context2vec*, all sentences consisting of more than 25 words are discarded.

[8]https://dumps.wikimedia.org/enwiki/20170601/

For post-training, dependency relations were derived using the Stanford Parser. To avoid the data sparseness problem, *dependency-words* were limited to nouns, verbs, adjectives, and adverbs. The number of training epochs in the post-training was set to one because our post-training aims to contextualize word embeddings that have been pre-trained. Hence, a long-time training does not have to be assumed. In the future, we plan to investigate the effects of the number of training epochs in post-training.

5.2 Evaluation Dataset

We used the following datasets in the evaluation.

LS-SE
: This is an official evaluation dataset in the lexical substitution task of SemEval-2007. For each target word, five annotators suggested up to three different substitutions. As the context, a sentence where a target word appears is provided. Every target has ten context sentences. The number of targets is 201 (types). Consequently, there are $2,010$ sets of target, candidates, and context sentences are available.

LS-CIC
: This is a large-scale dataset for a lexical substitution task. For $15,629$ target words, six annotators suggested up to five different substitutions under a context. Unlike LS-SE, three sentences are provided as context: a sentence containing the target word, its preceding sentence, and its following sentences.

CEFR-LP
: Our new dataset for lexical substitution, which is described in Section 4.

5.3 Evaluation Metrics

We used GAP (Kishida, 2005; Thater et al., 2009) as an evaluation metric. GAP is a commonly used metric to evaluate lexical substitution tasks. GAP calculates the ranking accuracy by considering the weight of correct examples:

$$p_i = \frac{\sum_{k=1}^{i} x_k}{i},$$

$$GAP = \frac{100 \sum_{i=1}^{n} I(x_i) p_i}{\sum_{i=1}^{n} I(y_i) \overline{y_i}},$$

where x_i and y_i represent the weight of the i-th substitution candidate ranked by an automatic method and by the ideal ranking, respectively. n represents the number of substitution candidates. $I(x)$ ($x \in \mathbb{N}$) is a binary function that returns 1 if $x \geq 1$. Otherwise, it returns 0. In this experiment, we regarded the number of annotators suggesting a substitution candidate under a given context as the weight of the candidate for LS-SE and LS-CIC. For CEFR-LP, we used the weight of the candidate that was computed based on the annotated labels as described in Section 4.2.

5.4 Baseline Method

We used the following baselines for comparison.

DMSE **(S_{maxc})**
: For *dependency-word* selection, S_{maxc} showing the highest performance is used herein. This is the best-performing model among the methods that generate contextualized word embeddings.

Context2vec
: This is the current state-of-the-art method among those proposed for lexical substitution. Note that this corresponds to the pre-trained model of our method.

ELMo
: We concatenate embeddings generated from three hidden layers in *ELMo* as contextualized word embeddings.[9]

DMSE and *ELMo* were trained using the same Wikipedia corpus as our method. These methods rank the substitution candidates in descending order of the cosine similarity between embeddings of the target and substitution candidate. For *context2vec*, the candidates are ranked in the same manner using our method based on Equation (1).

5.5 Ideal Selection of *Dependency-words*

The performance of our method depends on how *dependency-words* are selected. We simulate the performance when our method selects ideal *dependency-words* that maximize the GAP score. This selection method of *dependency-words* is denoted as $S_{\mathbf{best}}$.

[9] As a preliminary experiment, we compared methods to generate contextualized word embeddings. One used one of three layers of embeddings, one summed these embeddings, and one concatenated these embeddings. The results confirmed that concatenation performed best.

Model	LS-SE	LS-CIC	CEFR-LP
DMSE ($S_{\mathbf{maxc}}$)	49.3	46.5	71.1
ELMo	47.6	48.1	74.9
context2vec	51.1	50.0	75.3
context2vec + DMSE ($S_{\mathbf{maxc}}$)	52.2	50.9	75.5
context2vec + DMSE ($S_{\mathbf{tar}}$)	**52.3**	50.9	**75.6**
context2vec + DMSE ($S_{\mathbf{can}}$)	**52.3**	**51.0**	**75.6**
context2vec + DMSE ($S_{\mathbf{best}}$)	55.6	52.9	77.2

Table 6: GAP scores on LS-SE, LS-CIC and CEFR-LP datasets, where **bold** denotes the highest scores.

	A1	A2	B1	B2	C1 / C2
DMSE ($S_{\mathbf{maxc}}$)	67.9	75.2	65.7	74.9	72.7
context2vec	75.2	78.5	69.4	75.7	75.2
context2vec+DMSE ($S_{\mathbf{can}}$)	**75.3**	**78.9**	**69.9**	**76.2**	**75.9**

Table 7: GAP scores on different CEFR levels of target words in CEFR-LP

6 Results

Table 6 shows the GAP scores for LS-SE, LS-CIC and CEFR-LS datasets. Our method is denoted as *context2vec + DMSE* where the *dependency-word* selection method is represented in parenthesis as $S_{\mathbf{maxc}}$, $S_{\mathbf{tar}}$, or $S_{\mathbf{can}}$.

When using $S_{\mathbf{can}}$ for *dependency-word* selection, *context2vec + DMSE* outperformed *DMSE* by 3.0 points, 4.5 points, and 4.5 points for LS-SE, LS-CIC, and CEFR-LP, respectively. It even outperformed *context2vec*, the current state-of-the-art method, by 1.2 points, 1.0 points, and 0.3 points on these datasets, respectively. These results confirm the effectiveness of our method, which combines contextualized word embeddings and context embeddings to complement each other.

All *dependency-word* selection methods show fairly competitive performances, but $S_{\mathbf{can}}$ consistently achieved the highest GAP scores. Context embedding may be effective to select *dependency-words* rather than comparing contextualized word embeddings. The last row of Table 6 shows the performance of our method with S_{best} (*i.e.,* when the ideal *dependency-word* was selected). This best selection method outperformed 1.6 - 3.3 points higher than our method with $S_{\mathbf{can}}$, demonstrating the importance of *dependency-word* selection. In the future, we will improve the selection method.

CEFR-LP analyzes performances from the perspective of the CEFR levels of target words. Table 7 shows the GAP score of *DMSE* ($S_{\mathbf{maxc}}$), *context2vec*, and *context2vec+DMSE* ($S_{\mathbf{can}}$). Note that scores are *not* comparable across levels because the number of appropriate substitution candidates varies. Our method consistently outperforms *DMSE* ($S_{\mathbf{maxc}}$) and *context2vec*. Such an analysis is important when applying lexical substitution to educational applications.

Table 8 lists the results where each row shows a ranking of substitution candidates by compared methods. The annotated weights of each candidate are presented in parentheses. Here, the outputs of *context2vec+DMSE* ($S_{\mathbf{maxc}}$) to use the same *dependency-words* with *DMSE* ($S_{\mathbf{maxc}}$).

Inputs (1) and (2) show the cases where the meanings of polysemous target words (`go` and `tender`) are successfully captured by our method. It ranks `start` and `soft` first for each target, respectively. On the other hand, *DMSE* (S_{maxc}) failed to rank correct candidates higher although it referred to the same *dependency-words*. *Context2vec* also failed, but it used context embeddings. These results demonstrate that contextualized word embeddings and context embeddings complement each other. On Input (3), both *DMSE* (S_{maxc}) and our method failed while *context2vec* successfully rank the correct candidate (`grasp`) on top. This is caused by incorrect *dependency-word* selection. In Input (3), there are two major *dependency-words*, `sat` and `hands`. In this context, `hands` should be useful as a clue to iden-

Input (1)	To make these techniques work well , explain the basic concept and purpose and *get* it **going** with minimal briefing .
DMSE ($S_{\mathbf{maxc}}$)	try (0), move (1), proceed (1), leave (0), ...
context2vec	proceed (1), run (0), start (4), move (1), ...
context2vec+DMSE ($S_{\mathbf{maxc}}$)	start (4), proceed (1), move (1), run (0), ...
Input (2)	Rabbits often feed on young , **tender** perennial *growth* as it emerges in spring , or on young transplants .
DMSE ($S_{\mathbf{maxc}}$)	immature (0), young (0), great (1), soft (4), ...
context2vec	delicate (1), immature (0), soft (4), painful (0), ...
context2vec+DMSE ($S_{\mathbf{maxc}}$)	soft (4), delicate (1), immature (0), young (0), ...
Input (3)	A doctor *sat* in front of me and **held** my *hands* .
DMSE ($S_{\mathbf{maxc}}$)	put (0), lift (1), grasp (3), carry (0), ...
context2vec	grasp (3), carry (0), take (1), keep (0), ...
context2vec+DMSE ($S_{\mathbf{maxc}}$)	take (1), carry (0), keep (0), lift (1), ...

Table 8: Example outputs of each method. Target words in the input sentences are presented in **bold** and all of their *dependency-words* are presented in *italic*. Outputs are ranked lists of candidates, where the numbers in parentheses show candidates' weights. Our method ranks the appropriate candidates on top for the first two examples, but it failed on the last example due to incorrect *dependency-word* selection.

tify target word's sense but `sat` was mistakenly selected as the *dependency-word*. This result suggests that dependency types matter when selecting *dependency-words*, which we will tackle in the future.

7 Conclusion

Herein we proposed a method that combines *DMSE* and *context2vec* to simultaneously take advantage of contextualized word embeddings and context embeddings. The evaluation results on lexical substitution tasks confirm the effectiveness of our method, which outperforms the current state-of-the-art method. We also create a new evaluation set for lexical substitution tasks called CEFR-LP.

In the future, we will consider the dependency types in contextualized word embeddings for further improvements. Additionally, we plan to extend CEFR-LP to cover phrasal substitutions.

Acknowledgments

We thank Professor Christopher G. Haswell for his valuable comments and discussions. We also thank the anonymous reviewers for their valuable comments. This research was supported by the KDDI Foundation.

References

Kazuki Ashihara, Tomoyuki Kajiwara, Yuki Arase, and Satoru Uchida. 2018. Contextualized Word Representations for Multi-Sense Embedding. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation*, pages 28–36.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Learning Topic-Sensitive Word Representations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 441–447.

Kazuaki Kishida. 2005. Property of Average Precision and its Generalization: An Examination of Evaluation Indicator for Information Retrieval Experiments. *National Institute of Informatics Technical Reports*, pages 1–19.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What Substitutes Tell Us - Analysis of an "All-Words" Lexical Substitution Corpus. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 55–60.

Diana McCarthy. 2002. Lexical Substitution as a Task for WSD Evaluation. In *Proceedings of the ACL Workshop on Word Sense Disambiguation*, pages 109–115.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of the International Workshop on Semantic Evaluations*, pages 48–53.

Oren Melamud, Ido Dagan, and Jacob Goldberger. 2015. Modeling Word Meaning in Context with Substitute Vectors. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 472–482.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning*, pages 51–61.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceeding of the International Conference on Learning Representations*, pages 1–12.

Gustavo Henrique Paetzold and Lucia Specia. 2016. Unsupervised Lexical Simplification for Non-Native Speakers. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, pages 3761–3767.

Gustavo Henrique Paetzold and Lucia Specia. 2017. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, pages 549–593.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2227–2237.

Stephen Roller and Katrin Erk. 2016. PIC a Different Word: A Simple Model for Lexical Substitution in Context. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1121–1126.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, pages 2673–2681.

Stefan Thater, Universität Saarlandes, Georgiana Dinu, Universität Saarlandes, Manfred Pinkal, and Universität Saarlandes. 2009. Ranking Paraphrases in Context. In *Proceedings of the Workshop on Applied Textual Inference*, pages 44–47.

Satoru Uchida, Shohei Takada, and Yuki Arase. 2018. CEFR-based Lexical Simplification Dataset. *Proceedings of the International Conference on Language Resources and Evaluation*, pages 3254–3258.