BERT-based Lexical Substitution

Wangchunshu Zhou[1] * **Tao Ge**[2] **Ke Xu**[1] **Furu Wei**[2] **Ming Zhou**[2]
[1]Beihang University, Beijing, China
[2]Microsoft Research Asia, Beijing, China
zhouwangchunshu@buaa.edu.cn, kexu@nlsde.buaa.edu.cn
{tage, fuwei, mingzhou}@microsoft.com

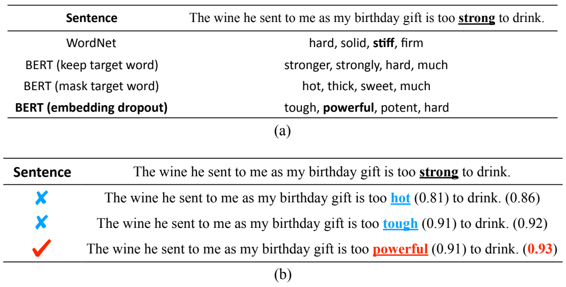
Abstract

Previous studies on lexical substitution tend to obtain substitute candidates by finding the target word's synonyms from lexical resources (e.g., WordNet) and then rank the candidates based on its contexts. These approaches have two limitations: **(1)** They are likely to overlook good substitute candidates that are not the synonyms of the target words in the lexical resources; **(2)** They fail to take into account the substitution's influence on the global context of the sentence.

To address these issues, we propose an end-to-end BERT-based lexical substitution approach which can propose and validate substitute candidates without using any annotated data or manually curated resources. Our approach first applies dropout to the target word's embedding for partially masking the word, allowing BERT to take balanced consideration of the target word's semantics and contexts for proposing substitute candidates, and then validates the candidates based on their substitution's influence on the global contextualized representation of the sentence. Experiments show our approach performs well in both proposing and ranking substitute candidates, achieving the state-of-the-art results in both LS07 and LS14 benchmarks.

1 Introduction

Lexical substitution (McCarthy and Navigli, 2007) aims to replace a target word in a sentence with a substitute word without changing the meaning of the sentence, which is useful for many Natural Language Processing (NLP) tasks like text simplification and paraphrase generation.



Sentence	The wine he sent to me as my birthday gift is too **strong** to drink.
WordNet	hard, solid, **stiff**, firm
BERT (keep target word)	stronger, strongly, hard, much
BERT (mask target word)	hot, thick, sweet, much
BERT (embedding dropout)	tough, **powerful**, potent, hard

(a)

Sentence	The wine he sent to me as my birthday gift is too **strong** to drink.
✗	The wine he sent to me as my birthday gift is too **hot** (0.81) to drink. (0.86)
✗	The wine he sent to me as my birthday gift is too **tough** (0.91) to drink. (0.92)
✓	The wine he sent to me as my birthday gift is too **powerful** (0.91) to drink. (**0.93**)

(b)

Figure 1: **(a)** WordNet and original BERT cannot propose the valid substitute *powerful* in their top-K results but applying target word embedding dropout enables BERT to propose it; **(b)** Undesirable substitutes (e.g., *hot*, *tough*) tend to change the contextualized representation of the sentence more than good substitutes (e.g., *powerful*). The numbers after the words are the cosine similarity of the words' contextualized vector to the original target words; while the numbers after the sentence are the similarity of the sentence's contextualized representation before and after the substitution, defined in Eq (2).

One main challenge in this task is proposing substitutes that not only are semantically consistent with the original target word and fits in the context but also preserve the sentence's meaning. Most previous approaches to this challenge first obtain substitute candidates by picking synonyms from manually curated lexical resources as candidates, and then rank them based on their appropriateness in context, or instead ranking all words in the vocabulary to avoid the usage of lexical resources. For example, knowledge-based lexical substitution systems (Yuret, 2007; Hassan et al., 2007) use pre-defined rules to score substitute candidates; vector space modeling approach (Erk and Padó, 2008; Dinu and Lapata, 2010; Thater et al., 2010; Apidianaki, 2016) uses distributional sparse vector representations based on the syntactic context; substitute vector approach (Yuret, 2012; Melamud et al., 2015b) comprises the potential fillers for the target word slot in that context; word/context embedding similarity approach (Melamud et al., 2015a; Roller and Erk,

[*] This work was done during the first author's internship at Microsoft Research Asia.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3368–3373
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

2016; Melamud et al., 2016) uses the similarity of word embeddings to rank substitute words; and supervised learning approaches (Biemann, 2013; Szarvas et al., 2013a,b; Hintz and Biemann, 2016) uses delexicalized features to rank substitute candidates. Although these approaches work well in some cases, they have two key limitations: **(1)** they rely heavily on lexical resources. While the resources can offer synonyms for substitution, they are not perfect and they are likely to overlook some good candidates, as Figure 1(a) shows. **(2)** most previous approaches only measure the substitution candidates' fitness given the context but they do not consider whether the substitution changes the sentence's meaning. Take Figure 1(b) as an example, although *tough* may fit in the context as well as *powerful*, it changes the contextualized representation of the sentence more than *powerful*. Therefore, it is not so good as *powerful* for the substitution.

To address the above issues, we propose a novel BERT-based lexical substitution approach, motivated by that BERT (Devlin et al., 2018) not only can predict the distribution of a masked target word conditioned on its bi-directional contexts but also can measure two sentences' contextualized representation's similarity. To propose substitute candidates for a target word in a sentence, we introduce a novel embedding dropout mechanism to partially mask the target word and use BERT to predict the word at the position. Compared to fully masking or keeping the target word, partially masking with embedding dropout allows BERT to take a balanced consideration of target word's semantics and its contexts, helping avoid generating substitute candidates that are either semantically inconsistent with the target word or unfit in the contexts, as Figure 1(a) shows. To validate a substitute candidate, we propose to evaluate a candidate's fitness based on the substitution's influence on the contextualized representation of the sentence, which avoids selecting a substitute that changes the sentence's meaning much, as Figure 1(b) illustrates. We conduct experiments on the official LS07 and LS14 benchmarks. The results show that our approach substantially outperforms previous approaches in both proposing and validating substitute candidates, achieving new state-of-the-art results in both datasets.

The contributions of our paper are as follows:

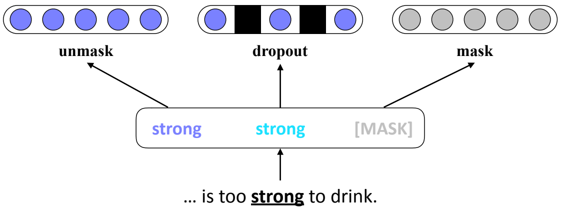


Figure 2: Unmasking, masking and partially masking the target word through target embedding dropout.

- We propose a BERT-based end-to-end lexical substitution approach without relying on any annotated data and external linguistic resources.

- Based on BERT, we introduce target word embedding dropout for helping substitute candidate proposal, and a substitute candidate validation method based on the substitution's influence on the global contexts.

- Our approach largely advances the state-of-the-art results of lexical substitution in both LS07 and LS14 benchmarks.

2 BERT-based Lexical Substitution

BERT (**B**idirectional **E**ncoder **R**epresentations from **T**ransformers) (Devlin et al., 2018) is a bidirectional transformer encoder (Vaswani et al., 2017) trained with the objective of masked language modeling and the next-sentence prediction task, which proves effective in various NLP tasks. In this section, we present how to effectively leverage BERT for lexical substitution.

2.1 Substitute Candidate Proposal

As BERT is a bi-directional language model trained by masking the target word, it can be used to propose a substitute candidate to reconstruct the sentence. In practice, however, if we mask the target word and let BERT predict the word at the position, BERT is very likely to generate candidates that are semantically different from the original target word although it fits in the context; on the other hand, if we do not mask the target word, approximately 99.99% of the predicted probability distribution will fall into the original target word, making it unreliable to choose the alternative candidates from the remaining 0.01% probability space, as Figure 1 shows.

For a trade-off between the two extreme cases, we propose to apply embedding dropout to partially mask the target word. It forces a portion of dimension of the target word's input embedding to zero, as illustrated in Figure 2. In this way, BERT can only receive vague information from the target word and thus has to consider other contexts to reconstruct the sentence, which improves substitute candidate proposal as Figure 1(a) shows.

Formally, for the target word x_k to be replaced in sentence $\boldsymbol{x} = (x_1, \cdots, \underline{x_k}, \cdots, x_L)$, we define $s_p(x'_k|\boldsymbol{x}, k)$ as the proposal score for choosing x'_k as the substitution for x_k:

$$s_p(x'_k|\boldsymbol{x}, k) = \log \frac{P(x'_k|\widetilde{\boldsymbol{x}}, k)}{1 - P(x_k|\widetilde{\boldsymbol{x}}, k)} \qquad (1)$$

where $P(x_k|\boldsymbol{x}, k)$ is the probability for the k^{th} word predicted by BERT given $\boldsymbol{x}$, and $\widetilde{\boldsymbol{x}}$ is the same with $\boldsymbol{x}$ except that its k^{th} position's word is partially masked with embedding dropout. The denominator is the probability of the prediction that is not x_k, normalizing $P(x'_k|\widetilde{\boldsymbol{x}}, k)$ against all the words in the vocabulary excluding x_k.

2.2 Substitute Candidate Validation

After we propose substitute candidates, we need to validate them because not all proposed candidates are appropriate. As Figure 1(b) shows, a proposed candidate (e.g., *tough*) may change the sentence's meaning. To avoid such cases, we propose to evaluate a candidate's fitness by comparing the sentence's contextualized representation before and after the substitution for validation.

Specifically, for a word x_i, we use the concatenation of its representations in top four layers in BERT as its contextualized representation. We denote the sentence after the substitution as $\boldsymbol{x}' = (x_1, \cdots, \underline{x'_k}, \cdots, x_L)$. The validation score for the substitution of x'_k is defined in Eq (2):

$$s_v(x'_k|\boldsymbol{x}, k) = \textsc{Sim}(\boldsymbol{x}, \boldsymbol{x}'; k) \qquad (2)$$

where $\textsc{Sim}(\boldsymbol{x}, \boldsymbol{x}'; k)$ is BERT's contextualized representation similarity of $\boldsymbol{x}$ and $\boldsymbol{x}'$, which is defined as follows:

$$\textsc{Sim}(\boldsymbol{x}, \boldsymbol{x}'; k) = \sum_{i}^{L} w_{i,k} \times \Lambda(\boldsymbol{h}(x_i|\boldsymbol{x}), \boldsymbol{h}(x'_i|\boldsymbol{x}'))$$

where $\boldsymbol{h}(x_i|\boldsymbol{x})$ is BERT's contextualized representation of the i^{th} token in the sentence $\boldsymbol{x}$ and $\Lambda(\boldsymbol{a}, \boldsymbol{b})$ is cosine similarity of vector $\boldsymbol{a}$ and $\boldsymbol{b}$. $w_{i,k}$ is the average self-attention score of all heads in all layers from i^{th} token to k^{th} position in $\boldsymbol{x}$, which is used for weighing each position based on its semantic dependency to x_k.

In this way, we can use $s_v(x'_k|\boldsymbol{x}, k)$ to measure the influence of the substitution of $x_k \rightarrow x'_k$ on the semantics of the sentence. The undesirable substitute candidates like *hot* and *tough* in Figure 1(b) will get a lower s_v and thus fail in ranking, while appropriate candidates like *powerful* will have a high s_v and will be preferred.

In practice, we consider both the proposal score s_p in Eq (1) and the validation score s_v in Eq (2) for overall recommendation for a candidate:

$$s(x'_k|\boldsymbol{x}, k) = s_v(x'_k|\boldsymbol{x}, k) + \alpha \times s_p(x'_k|\boldsymbol{x}, k) \quad (3)$$

where α is the weight for the proposal score.

3 Experiments

3.1 Experimental Setting

We evaluate our approach on the SemEval 2007 dataset (McCarthy and Navigli, 2007) (denoted as LS07), and the CoinCo dataset (Kremer et al., 2014) (denoted as LS14), benchmark datasets which are the most widely used datasets for lexical substitution evaluation. LS07 consists of 201 target word types, each of which has 10 instances in different contexts (i.e., sentences); while LS14 provides the same kind of data as LS07 but is much larger – with 4,255 target word types in over 15K sentences.

We use official evaluation metrics *best*, *best-mode*, *oot*, *oot-mode* in SemEval 2007 task as well as Precision@1 as our evaluation metrics. Among them, *best*, *best-mode* and Precision@1 evaluate the quality of the best predictions while *oot* (*out-of-ten*) and *oot-mode* evaluate the coverage of the gold substitutes in 10-best predictions.

We use uncased BERT large model in Devlin et al. (2018) in our experiments. We use LS07 trial set as our development set for tuning the hyperparameters in our model. Empirically, we set the dropout ratio of the target word's embedding to 0.3 and set the weight α in Eq (3) to 0.01. For each test instance, we propose 50 candidates using the approach in Section 2.1 and validate and rank them by Eq (3). As the embedding dropout introduces randomness to the final results, we repeat our experiments 5 times and report average scores with standard deviation.

3.2 Experimental Results

Table 1 shows the results of our approaches as well as the state-of-the-art approaches in LS07 and LS14 benchmarks. Our approach substantially outperforms all previous approaches in both

Method	Resource	best	best-m	oot	oot-m	P@1
			LS07			
our approach	None	**20.3**±0.02	**34.2**±0.02	**55.4**±0.03	**68.4**±0.02	**51.1**±0.02
substitute vector (Melamud et al., 2015b)	None	12.7	21.7	36.4	52.0	-
balAddCos (Melamud et al., 2015a)	None	8.1	13.4	27.4	39.1	13.4
transfer learning (Hintz and Biemann, 2016)	WordNet	17.2	-	48.8	-	-
supervised learning (Szarvas et al., 2013b)	WordNet	15.9	-	48.8	-	40.8
KU (knowledge-based) (Yuret, 2007)	multiple resources	12.9	20.7	46.2	61.3	-
UNT (knowledge-based) (Hassan et al., 2007)	multiple resources	12.8	20.7	49.2	66.3	-
			LS14			
our approach	None	**14.5**±0.01	**33.9**±0.02	**45.9**±0.02	**69.9**±0.02	**56.3**±0.01
substitute vector (Melamud et al., 2015b)	None	8.1	17.4	26.7	46.2	-
balAddCos (Melamud et al., 2015a)	None	5.6	11.9	20.0	33.3	11.8

Table 1: Results on LS07 and LS14. For all the metrics, the higher, the better. For substitution vector and balAddCos, they use all the words in the vocabulary as the substitution candidates.

Method	best	best-m	oot	oot-m	P@1
		LS07			
our approach	**20.3**	**34.2**	**55.4**	**68.4**	**51.1**
- w/o s_p (Keep)	18.9	32.6	51.7	63.5	48.6
- w/o s_p (Mask)	16.2	27.5	46.4	57.9	43.3
- w/o s_p (WordNet)	15.9	27.1	45.9	57.1	42.8
- w/o s_v	12.1	20.2	40.8	56.9	13.1
BERT (Keep)	9.2	16.3	37.3	52.2	9.2
BERT (Mask)	8.6	14.2	33.2	48.9	5.7
		LS14			
our approach	**14.5**	**33.9**	**45.9**	**69.9**	**56.3**
- w/o s_p (Keep)	13.7	31.4	41.3	63.5	53.1
- w/o s_p (Mask)	11.3	26.7	36.2	59.1	47.1
- w/o s_p (WordNet)	11.0	26.3	35.9	58.7	46.3
- w/o s_v	9.1	19.7	33.5	56.9	14.3
BERT (Keep)	8.3	17.2	31.1	54.4	11.2
BERT (Mask)	7.6	15.4	38.5	51.3	7.6

Table 2: Ablation study results of our approach. *BERT (Keep/Mask)* are the baselines that uses BERT unmasking/masking the target word to propose candidates and rank by the proposal scores. Remember that **our approach** is a linear combination of proposal score s_p and validation score s_v, as in Eq (3). In the baselines "w/o s_p", we alternatively use *BERT (Keep)*, *BERT (Mask)* or WordNet to propose candidates.

Method	LS07	LS14
our approach	**60.5**	**57.6**
- w/o s_v	55.3	52.2
- w/o s_p	58.3	54.8
context2vec (Melamud et al., 2016)	56.0	47.9
substitute vector (Melamud et al., 2015b)	55.1	50.2
addcos (Melamud et al., 2015a)	52.9	48.3
PIC (Roller and Erk, 2016)	52.4	48.3
vector space modeling (Kremer et al., 2014)	52.5	47.8
transfer learning (Hintz and Biemann, 2016)	51.9	-
supervised learning (Kremer et al., 2014)	55.0	-
BERT (word similarity)	55.2	52.1

Table 3: GAP scores in the substitute ranking subtask. Note that for the baseline w/o s_p, we do not need to propose candidates using BERT like Table 2 since candidates are given in advance in the ranking subtask. *BERT (word similarity)* ranks candidates by the cosine similarity of BERT contextualized representations of the original target word and a substitute candidate. We do not compare to Apidianaki (2016) as it only evaluates on a sample of the test data in a different setting.

benchmarks, even those trained through supervised learning with external resources (Szarvas et al., 2013b), in terms of all the five metrics. Though our approach introduces randomness due to the embedding dropout, no large fluctuation is observed in our results.

For understanding the improvement, we conduct an ablation test and show the result in Table 2. According to Table 2, we observe that the original BERT cannot perform as well as the previous state-of-the-art approaches by its own. Applying embedding dropout to BERT improves the model, allowing it to achieve 13.1% and 14.3% P@1 in LS07 and LS14 respectively. When we further add our candidate valuation method in Section 2.2 to validate the candidates, its performance is significantly improved. Furthermore, it is clear that our substitute candidate proposal method is much better than WordNet for candidate proposal when we compare our approach to the -w/o s_p (WordNet) baseline where candidates are obtained by WordNet and validated by our validation approach.

Also, we evaluate our approach in the substitute ranking subtask of LS07 and LS14. In the ranking subtask, a system does not need to propose candidates by itself; instead, the substitute candidates for each test instance are given in advance, either from lexical resources (e.g. wordnet) or pooled substitutes. Following prior work, we use GAP score (Kishida, 2005) for evaluation in the subtask, which is a variant of MAP (Mean Average Precision). According to Table 3, we observe that both our proposal score s_p and validation score s_v contribute to the improvement, allowing our approach to outperform previous state-of-the-art approaches, even with the same substitute candidates.

By comparing our approach without s_p to the BERT baseline approach *BERT (word similarity)* in Table 3, we confirm that the comparison of sentence-level contextualized representations before and after the substitution is more effective and reliable than the word-level comparison for lexical substitution. This is because some changes in sentence's meaning after the substitution can be better captured by the sentence-level analysis, just as the example in Figure 1(b) illustrates.

4 Conclusion

In our work, we propose an end-to-end lexical substitution approach based on BERT, which can propose and validate substitute candidates without using any annotated data and manually curated resources. Experiments in LS07 and LS14 benchmark datasets show that our proposed embedding dropout for partially masking the target word is helpful for BERT to propose substitute candidates, and that analyzing a sentence's contextualized representation before and after the substitution can largely improve the results of lexical substitution.

Acknowledgments

We thank the anonymous reviewers for their valuable comments.

References

Marianna Apidianaki. 2016. Vector-space models for ppdb paraphrase ranking in context. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2028–2034.

Chris Biemann. 2013. Creating a system for lexical substitutions from scratch using crowdsourcing. *Language Resources and Evaluation*, 47(1):97–122.

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*.

Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1162–1172. Association for Computational Linguistics.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 897–906. Association for Computational Linguistics.

Samer Hassan, Andras Csomai, Carmen Banea, Ravi Sinha, and Rada Mihalcea. 2007. Unt: Subfinder: Combining knowledge sources for automatic lexical substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 410–413. Association for Computational Linguistics.

Gerold Hintz and Chris Biemann. 2016. Language transfer learning for supervised lexical substitution. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 118–129.

Kazuaki Kishida. 2005. Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments.

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 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 48–53, Stroudsburg, PA, USA. Association for Computational Linguistics.

Oren Melamud, Ido Dagan, and Jacob Goldberger. 2015a. Modeling word meaning in context with substitute vectors. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 472–482.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61.

Oren Melamud, Omer Levy, and Ido Dagan. 2015b. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7.

Stephen Roller and Katrin Erk. 2016. 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*, pages 1121–1126.

György Szarvas, Chris Biemann, and Iryna Gurevych. 2013a. Supervised all-words lexical substitution using delexicalized features. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1131–1141.

György Szarvas, Róbert Busa-Fekete, and Eyke Hüllermeier. 2013b. Learning to rank lexical substitutions. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1926–1932.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 948–957. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Deniz Yuret. 2007. Ku: Word sense disambiguation by substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 207–213, Stroudsburg, PA, USA. Association for Computational Linguistics.

Deniz Yuret. 2012. Fastsubs: An efficient and exact procedure for finding the most likely lexical substitutes based on an n-gram language model. *IEEE Signal Processing Letters*, 19(11):725–728.