LexSubCon: Integrating Knowledge from Lexical Resources into Contextual Embeddings for Lexical Substitution

George Michalopoulos, Ian McKillop, Alexander Wong, Helen Chen
University of Waterloo,
Waterloo, Canada
{gmichalo, ian, alexander.wong, helen.chen}@uwaterloo.ca

Abstract

Lexical substitution is the task of generating meaningful substitutes for a word in a given textual context. Contextual word embedding models have achieved state-of-the-art results in the lexical substitution task by relying on contextual information extracted from the replaced word within the sentence. However, such models do not take into account structured knowledge that exists in external lexical databases.

We introduce LexSubCon, an end-to-end lexical substitution framework based on contextual embedding models that can identify highly-accurate substitute candidates. This is achieved by combining contextual information with knowledge from structured lexical resources. Our approach involves: (i) introducing a novel mix-up embedding strategy to the target word's embedding through linearly interpolating the pair of the target input embedding and the average embedding of its probable synonyms; (ii) considering the similarity of the sentence-definition embeddings of the target word and its proposed candidates; and, (iii) calculating the effect of each substitution on the semantics of the sentence through a fine-tuned sentence similarity model. Our experiments show that LexSubCon outperforms previous state-of-the-art methods by at least 2% over all the official lexical substitution metrics on LS07 and CoInCo benchmark datasets that are widely used for lexical substitution tasks.

1 Introduction

Lexical Substitution (McCarthy and Navigli, 2007) is the task of generating appropriate words which can replace a target word in a given sentence without changing the sentence's meaning. The increased research interest in Lexical Substitution is due to its utility in various Natural Language Processing (NLP) fields including data augmentation, paraphrase generation and semantic text similarity.

Contextual word embedding models (such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019)) have achieved state-of-art results in many NLP tasks. These models are usually pre-trained on massive corpora and the resulting context-sensitive embeddings are used in different downstream tasks (Howard and Ruder, 2018). Zhou et al. (2019) have achieved state-of-the-art results on the lexical substitution task by improving the BERT's standard procedure of the masked language modeling task. However, the current state-of-the-art contextual models have yet to incorporate structured knowledge that exists in external lexical database into their prediction process. These lexical resources could boost the model's performance by providing additional information such as the definitions of the target and candidate words (in order to ensure that the candidate word is semantically similar to the target word and not only appropriate for the sentence's context) or by enriching the proposed candidate word list so it will not only be based on the vocabulary of the contextual model.

In this paper, we present and publicly release[1] a novel framework for the lexical substitution task. Specifically, (i) we are the first, to the best of our knowledge, to propose a novel mix-up embedding strategy that outperforms the previous state-of-the-art strategy of word embedding dropout for the input embedding of the target word in a contextual model (Zhou et al., 2019) for the task of predicting accurate candidate words; (ii) we propose the combined usage of features from contextual embedding models and external lexical knowledge bases in order to determine the most appropriate substitution words without modifying the meaning of the original sentence, such as introducing a new gloss (definition) similarity metric which calculates the similarity of the sentence-definition embeddings of the target word and its proposed candidates; (iii) we generate a highly accurate fine–tuned sentence similarity model by taking advantage of popular data augmentation techniques (such

[1] https://github.com/gmichalo/LexSubCon

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1226 - 1236
May 22-27, 2022

as back translation), for calculating the effect of each candidate word in the semantics of the original sentence; and, (iv) finally, we show that LexSubCon achieves state-of-the-art results on two popular benchmark lexical substitution datasets (McCarthy and Navigli, 2007; Kremer et al., 2014).

2 Related Work

The lexical substitution task consists of two subtasks: (i) generating a set of meaning preserving substitute candidates for the target word and (ii) appropriately ranking the words of the set by their ability to preserve the meaning of the initial sentence (Giuliano et al., 2007; Martinez et al., 2007). However, lexical substitution models can also be tested in a "simpler" problem where the set of substitute candidates is composed of human-suggested words and the task is to accurately rank the substitute words that are provided (Erk and Padó, 2010).

The authors in (Melamud et al., 2015b) proposed the use of a word2vec model which utilizes word and context embeddings to represent the target word in a given context. Their model ranked the candidate substitutions by measuring their embedding similarity. In (Melamud et al., 2016) the context2vec model was introduced where the context representation of the word was calculated by combining the output of two bidirectional LSTM models using a feed-forward neural network.

Peters et al. (2018) introduced contextualized word embeddings in a bidirectional language model (ELMo). This allowed the model to change the embedding of a word based on its imputed meaning which is derived from the surrounding context. Subsequently, Devlin et al. (2019) proposed the Bidirectional Encoder Representations from Transformers (BERT) which uses bidirectional transformers (Vaswani et al., 2017) to create context-dependent representations. The authors in (Garí Soler et al., 2019) used ELMo in the lexical substitution task by calculating the cosine similarity between the ELMo embedding of the target word and all the candidate substitutes. In addition, Zhou et al. (2019) achieved state-of-the-art results on the lexical substitution task by applying a dropout embedding policy to the target word embedding and by taking into account the similarity between the initial contextualized representation of the context words and their representation after replacing the target word by one of the possible candidate words. An analysis of state-of-the-art contextual model on the lexical substitution task was presented in (Arefyev et al., 2020).

Finally, external knowledge from knowledge bases has been used to enhance the performance of deep learning models. Sense-BERT (Levine et al., 2020) was pre-trained to predict the semantic class of each word by incorporating lexical semantics (from the lexical database WordNet (Miller, 1995)) into the model's pre-training objective. Furthermore, Faruqui et al. (2015) and Bahdanau et al. (2017) used external knowledge (namely WordNet) in order to enhance word embeddings and to create more accurate representations of rare words.

3 LexSubCon Framework

In the lexical substitution task, a model aims to firstly generate a set of candidate substitutions for each target word and secondly to create an appropriate ranking of the elements of the candidate set. In addition, there are two main conditions for a lexical substitute model to satisfy: (i) to be semantically similar to the target word and (ii) and to be compatible with the given context (sentence) (Melamud et al., 2015b). We present the LexSubCon framework which achieves state of the art results on the lexical substitution task by combining contextual information with knowledge from structured external lexical resources.

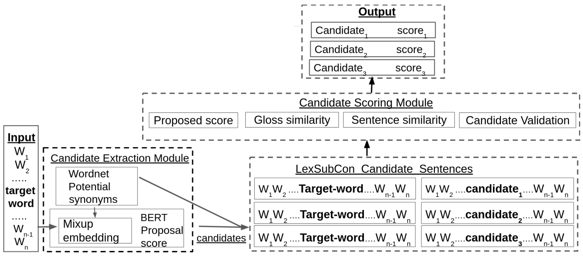
Output
Candidate$_1$ score$_1$
Candidate$_2$ score$_2$
Candidate$_3$ score$_3$
Candidate Scoring Module
Proposed score
Gloss similarity
Sentence similarity
Candidate Validation
Input
W$_1$
W$_2$
.....
target word
.....
W$_{n-1}$
W$_n$
Candidate Extraction Module
Wordnet Potential synonyms
Mixup embedding
BERT Proposal score
candidates
LexSubCon Candidate Sentences
W$_1$W$_2$ ....Target-word....W$_{n-1}$W$_n$
W$_1$W$_2$ ....candidate$_1$....W$_{n-1}$W$_n$
W$_1$W$_2$ ....Target-word....W$_{n-1}$W$_n$
W$_1$W$_2$....candidate$_2$....W$_{n-1}$W$_n$
W$_1$W$_2$ ....Target-word....W$_{n-1}$W$_n$
W$_1$W$_2$....candidate$_3$....W$_{n-1}$W$_n$

Figure 1: LexSubCon framework

The architecture of LexSubCon is depicted in Figure 1. The key characteristic of LexSubCon is its capability of unifying different substitution criteria such as contextualized representation, definition and sentence similarity in a single framework in order to accurately identify suitable candidates for the target words in a specific context (sentence).

3.1 Proposed Score: Mix-Up Embedding Strategy

The standard BERT architecture (Devlin et al., 2019) can be used in the lexical substitution task by masking the target word and letting the model to propose appropriate substitute candidates that preserve the initial meaning of the sentence. Zhou et al. (2019) argued that applying embedding dropout to partially mask the target word is a better alternative than masking the whole word. This is because the model may generate candidates that are semantically different but appropriate for the context of the initial sentence. Their experiments showed that this policy is indeed more beneficial than completely masking, or not masking, the target word.

However, in this paper we demonstrate that a mix-up embedding strategy can yield even better results. The main disadvantage of dropout embedding is that it sets random positions in the embedding vector of the target words to zero. We propose that by using external knowledge, we can obtain probable synonyms of the target word and use that knowledge in a mix-up scenario (Zhang et al., 2018) through linearly interpolating the pair of the target input embedding and the average embedding of its synonyms. This allows the model to generate a new synthetic input embedding by repositioning the target embedding around the neighborhood of the embedding of its synonyms. In order to obtain appropriate synonyms we use WordNet (Miller, 1995) which is an extensive lexical database where words are grouped into sets of synonyms (synsets). In our experiments, the best performance was achieved when the list of synonyms was extracted from the complete set of synsets for each word as it minimizes the chance of having a synonym set that only includes the target word itself.

Finally, we use a mix-up strategy to calculate a new input embedding for the target word X'_{target} as shown in equation 1:

$$X^{'}_{target} = \lambda X_{target} + (1-\lambda)\overline{X}_{synonyms} \quad (1)$$

Where X_{target} is the initial input embedding of the target word, $\overline{X}_{synonyms}$ is the average embedding of all the synonyms. It should be noted that Wordnet does not contain information about some words, such as pronouns, conjunctions, or nouns that are not commonly used in the English vocabulary. To address this limitation, whenever a target word cannot be found in the WordNet database, we replace the mix-up strategy by injecting Gaussian noise to the input embedding of the target word. This produces a similar effect as the mix-up strategy since the target embedding is re-positioned around itself in the embedding space (equation 2):

$$X^{'}_{target} = X_{target} + e \quad (2)$$

where e is a Gaussian noise vector with components $e_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

We use the BERT architecture to calculate the proposal score for each candidate. The input embedding vectors pass through multiple attention-based transformer layers where each layer produces a contextualized embedding of each token. For each target word x_t, the model outputs a score vector $y_t \in \mathbb{R}^D$, where D is the length of the model's vocabulary. We calculate the proposal score s_p for each candidate word x_c, using the score vector y_t of the BERT's language modeling process, as the probability for the BERT model to propose the word x_c over all the candidates words x'_c when the target word's sentence is provided as input to it:

$$s_p(x_c) = \frac{exp(y_t[x_c])}{\sum_{x'_c} exp(y_t[x'_c])} \quad (3)$$

3.2 Gloss-Sentence Similarity Score

In the previous section, we analyzed our model which ranks candidate substitute words by calculating their individual proposal scores. However, Zhou et al. (2019) and Arefyev et al. (2020) showed that the proposal score does not provide sufficient information about whether the substitute words will modify the sentence's meaning. Thus, in this section, we present a new metric which ranks the candidate words by considering the gloss (a dictionary-style definition) of each word. By extracting the appropriate information from the Wordnet database, a list of potential glosses is created for each target or candidate word. In addition, we can determine the most appropriate gloss based on the word and its specific context (sentence) by taking advantage of recent fine-tuned contextual models that have achieved state-of-the-art results in the Word Sense Disambiguation (WSD) task (Huang et al., 2019). As the glosses are sentences (sequence of words) they can be represented in a semantic space through a sentence embedding generating model. A ranking of each candidate word is produced by calculating the cosine similarity between the gloss sentence embedding of the target word and the gloss sentence embedding of each candidate word.

There are many methods for generating sentence embeddings, such as calculating the weighted average of its word embeddings (Arora et al., 2017). We select the sentence embeddings of the stsb-roberta-large model in (Reimers and Gurevych,

2019) which has been shown to outperform other state-of-the-art sentence embeddings methods.

Given a sentence s, a target word x_t and a candidate word x_c, our model first identifies the most appropriate gloss g_t for the target word given its context. After replacing the target word with the candidate x_c to create a new sentence s', the most appropriate gloss g_c for the candidate word is also determined. A gloss-similarity score s_g for each candidate is then calculated as the cosine similarity between the two glosses-sentences embeddings.

$$s_g(x_c) = cos(g_t, g_c) \tag{4}$$

3.3 Sentence Similarity Score

We also chose to calculate the effect of each substitution in the semantics of the original sentence by calculating the semantic textual similarity between the original sentence s and the updated sentence s' (a sentence where we have replaced the target word with one of its substitutions).

In order to accurately calculate a similarity score between s and s', we fine-tune a semantic textual similarity model based on the stsb-roberta-large model (Reimers and Gurevych, 2019) by using the training portion of the dataset in order to create pairs of sentences between the original sentence and an updated sentence where we have substitute the target word with one of its proposed candidates. Using the methods that we described in section 3.2, we can identify the most appropriate synset (from WordNet) for each target word and create a new pair of sentences between the original sentence and an updated sentence where we have updated the target word with the synonyms of the previous mentioned synset. However, due to the limited size of the training dataset, our model is still not provided with enough training data in order to be fully fine-tuned.

This is the reason why we employ a data augmentation technique in order to produce the examples needed for this task. Specifically, we create a back-translation mechanism in order to generate artificial training data. Back-translation or round-trip translation is the process of translating text into another language (forward translation) and then translating back again into the original language (back translation) (Aiken and Park, 2010). Back-translation has been used in different tasks in order to increase the size of training data (Sennrich et al., 2016; Aroyehun and Gelbukh, 2018). In our case, we provide to the back-translation module the initial sentence s and it produces a sightly different 'updated' sentence s'_u. For the s'_u sentences that still contain the target word we can create pair of sentences between the s'_u and an alternative version of the s'_u sentence (s''_u) where the target word is substituted with one of the candidate words or synonyms that we mentioned in the above paragraph. The main disadvantage of this techniques is that it may return the same initial sentence without any changes. In this case, we add a second translation level where the initial sentence is translated into two different languages before being translated back.

3.4 Candidate Validation Score

In our experiments we have also included the substitute candidate validation metric from (Zhou et al., 2019) as it has been shown to have a positive affect on the performance of a lexical substitution model. The substitute candidate validation metric is represented as the weighted sum of the cosine similarities between the contextual representation of each token in the initial and in the updated sentence where the weight of the cosine similarity of the token i is calculated as the average self-attention score of all heads in all layers from the token of the target word to token i. As mentioned in (Zhou et al., 2019), this metric evaluates the influence of the substitution on the semantic of the sentence.

Finally, LexSubCon uses a linear combination of the above mentioned features to calculate the final score for each candidate word.

3.5 Candidate Extraction

The candidates for each target word are extracted using the external lexical resource of WordNet and the BERT-based lexical substitution approach where the model provides probabilities for each candidate based on the context (sentence). We create a list of candidates based on the synonyms, the hypernyms, and hyponyms of each target word that could be identified in WordNet. In addition, we include in the list the candidate words with the higher probability that could be identified using the mix-up strategy that we described in section 3.1. We chose to include candidates from WordNet because we do not want our model to only include candidates words from the BERT vocabulary and we also include candidates words from a BERT-based model because target words may not be included in WordNet.

4 Experiments

4.1 Dataset

We evaluate LexSubCon on the English datasets SemEval 2007 (LS07)[2] (McCarthy and Navigli, 2007) and Concepts-In-Context (CoInCo)[3] (Kremer et al., 2014) which are the most widely used datasets for the evaluation of lexical substitution models. (i) The LS07 dataset is split into 300 train and 1710 test sentences where for each of the 201 target words there are 10 sentences (extracted from http://corpus.leeds.ac.uk/internet.html). The gold standard was based on manual annotation where annotators provided up to 3 possible substitutes. (ii) The CoInCo dataset consists of over 15K target word instances (based on texts provided in the Open American National Corpus) where 35% are training and 65% are testing data. Each annotator provided at least 6 substitutes for each target word. Our experiments with all datasets are consistent with their intended use, as they were created for research purposes. We manually investigate the existence of information that names individuals or offensive content, however, we did not find any indication of either of them.

In order to have a fair comparison with the previous state-of-the-art models, for both datasets we used their processed versions as used in (Melamud et al., 2015b, 2016).

4.2 Experimental Setup

LexSubCon was evaluated in the following variations of the lexical substitution tasks:

All-ranking task: In this task no substitution candidates are provided. We use the official metrics that the organizers provided in the original lexical substitution task of SemEval-2007 [4]. These were *best* and *best-mode* which validate the quality of the model's best prediction and both *oot* (out-of-ten) and *oot-mode* to evaluate the coverage of the gold substitute candidate list by the 10-top predictions. We also use $Precision@1$ to have a complete comparison with the model in (Zhou et al., 2019).

Candidate ranking task: In this task the list of candidates are provided and the goal of the model is to rank all the candidate words. For the candidate ranking task we follow the policy of previous works and construct the candidate list by merging all the substitutions of the target lemma and POS tag over the whole dataset. For measuring the performance of the model we use the GAP score (Kishida, 2005)[5] which is a variant of the MAP (Mean Average Precision). Following (Melamud et al., 2015b), we discard all multi-words from the gold substitutes list and remove the instances that were left with no gold substitutes.

We use the uncased BERT large model (Devlin et al., 2019) for the calculation of the proposal score and candidate validation score. For the identification of the most appropriate glosses we employ the pre-trained model in (Huang et al., 2019) which has achieved the state-of-the-art results in the Word Sense Disambiguation (WSD) task. Finally, the sentence-similarity metric is computed by finedtuning the stsb-roberta-large model in (Reimers and Gurevych, 2019) and by employing the OPUS-MT models (Tiedemann and Thottingal, 2020) (namely opus-mt-en-romance, opus-mt-fr-es and opus-mt-romance-en) for the creation of the back-translated sentences.

We use the LS07 trial set for training the sentence similarity metric model (for 4 epochs) and for fine-tuning the parameters of our framework based on the *best* score. Empirically, the λ parameter of the mix-up strategy was set to 0.25 and the weights to $0.05, 0.05, 1, 0.5$ for the proposal score, gloss-sentence similarity score, sentence similarity score and candidate validation score respectively (with the search space for all the parameters being $[0, 1]$[6]). Finally, for the Gaussian noise we choose a mean value of 0 and standard deviation 0.01. We propose 30 candidates for each target word. In order to achieve more robust results, we run LexSubCon on five different (random) seeds and we provide the average scores and standard deviation. All the contextual models are implemented using the transformers library (Wolf et al., 2019) on PyTorch 1.7.1. All experiments are executed on a Tesla K80 GPU with 64 GB of system RAM on Ubuntu 18.04.5 LTS. It should be noted that LexSubCon contains 1136209468 parameters.

4.3 Lexical Substitution Model Comparison

To enable direct comparison and to isolate gains due to improvements solely on the post-processing strategy that each model uses (which has the potential to change its performance (Arefyev et al., 2020)), we opt to reproduce and use the same strat-

[2] license: https://tinyurl.com/semeval-license

[3] license: CC-BY-3.0-US

[4] www.dianamccarthy.co.uk/files/task10data.tar.gz

[5] https://tinyurl.com/gap-measure

[6] As we only had four weight parameters, the identification of the best combination was finished in less than half an hour.

Method	best	best-m	oot	oot-m	P@1
			LS07 dataset		
LexSubCon	**21.1** ±0.03	**35.5**±0.07	**51.3** ±0.05	**68.6**±0.05	**51.7**±0.03
Bert$_{s_p,s_u}$*	12.8 ±0.02	22.1±0.03	43.9±0.01	59.7±0.02	31.7± 0.02
Transfer learning	17.2	-	48.8	-	-
Substitute vector	12.7	21.7	36.4	52.0	-
Addcos	8.1	13.4	27.4	39.1	-
Supervised learning	15.9	-	48.8	-	40.8
KU	12.9	20.7	46.2	61.3	-
UNT	12.8	20.7	49.2	66.3	-
			CoInCo dataset		
LexSubCon	**14.0** ± 0.02	**29.7** ± 0.03	**38.0** ± 0.03	**59.2** ± 0.04	**50.5** ± 0.02
Bert$_{s_p,s_u}$ *	11.8 ± 0.02	24.2 ± 0.02	36.0 ± 0.02	56.8 ± 0.02	43.5 ± 0.02
Substitute vector	8.1	17.4	26.7	46.2	-
Addcos	5.6	11.9	20.0	33.8	-

Table 1: Results of mean ± standard deviation of five runs from our implementation of LexSubCon and Bert$_{s_p,s_u}$(Zhou et al., 2019). We also provide the performance of previous state-of-the-art models. Transfer learning (Hintz and Biemann, 2016), Substitute vector (Melamud et al., 2015a), Addcos (Melamud et al., 2015b), Supervised learning (Szarvas et al., 2013b), KU (Yuret, 2007), UNT (Hassan et al., 2007). Best values are **bolded**.

egy for the tokenization of the target words from Bert$_{s_p,s_u}$ (Zhou et al., 2019). We focus our comparison on Bert$_{s_p,s_u}$ as it has achieved impressive state of the art results on both benchmark datasets[7].

The results of LexSubCon and the previous state-of-the art results in both LS07 and CoInCo benchmark datasets are presented in Table 1. LexSubCon outperformed the previous methods across all metrics in the LS07 and the CoInCo datasets due to the fact that all features have a positive contribution on its performance (see ablation details in section 4.5) as they encourage LexSubCon to take into consideration different substitution criteria. The standard deviation of the results of LexSubCon is not zero due to the fine-tuning process of the sentence similarity model. However, the results indicate that there are no large fluctuations. LexSubCon and our implementation of Bert$_{s_p,s_u}$ had a running time of 74k and 30k for LS07 and 580K and 266K seconds for the CoInCo dataset respectively.

4.4 Mix-Up Strategy Evaluation

In order to evaluate the mix-up strategy for the input embedding of the proposal model, we study the effect of different input embedding policies. The results of this study are listed in Table 2. It can be observed that even the simpler strategy of injecting Gaussian noise to the input embedding outperformed the standard policy of masking the input word. These results indicate that a contextual model needs information from the embedding of the target word in order to predict accurate candidates but it may over-rely on this information when it is provided with an intact input embedding. Fur-

Policy	best	best-m	oot	oot-m	P@1
		LS07			
Mix.	**16.3**	**27.6**	**45.6**	**62.4**	**40.8**
Gaus.	15.4	25.1	44.3	61.4	38.9
Drop.	15.5	25.6	44.3	61.2	38.8
Mask	10.4	16.4	35.5	48.6	27.0
Keep	15.5	25.4	44.4	61.4	39.2
		CoInCo			
Mix.	**11.3**	**23.8**	**33.6**	**54.4**	**41.3**
Gaus.	10.8	22.6	33.0	54.4	39.7
Drop.	10.8	22.5	32.9	54.2	39.5
Mask	8.6	17.5	28.9	46.6	31.7
Keep	10.8	22.6	33.0	54.3	39.7

Table 2: Comparison of different strategies for modifying the input embedding of the proposal model. *Mix.* is the mix-up strategy that we proposed, *Gaus.* is the Gaussian noise strategy, *Drop.* is the dropout embedding strategy in (Zhou et al., 2019), *Mask* is the strategy of masking the target word and *Keep* is the strategy of unmasking the target word in the input of the proposal model. Best values are **bolded**.

[7] Note that the method proposed by (Zhou et al., 2019) was implemented to the best of our abilities to be as faithful to the original work as possible using elements of code that the method's authors kindly provided upon request. However, the authors could not make the complete original code available to us.

thermore, the mix-up strategy outperformed all the other policies and specifically the dropout embedding strategy (Zhou et al., 2019) as the mix-up strategy re-positions the target embedding around the neighborhood of the embedding of its synonyms and it does not erase a part of the embedding that the model can learn from.

4.5 Ablation Study

Method	best	best-m	oot	oot-m	P@1
		LS07			
LexS	**21.1**	**35.5**	**51.3**	**68.6**	**51.7**
-w Pr.	20.1	32.6	50.8	68.1	50.6
-w Gl.	19.9	33.7	50.4	67.6	48.6
-w Sen.	20.7	34.9	50.9	68.2	50.6
-w Val.	18.8	31.7	47.8	64.9	46.6
Pr.	16.3	27.6	45.6	62.4	40.8
Gl.	12.4	19.5	40.5	55.0	32.7
Sen.	16.7	28.3	45.3	62.0	40.7
Val.	18.6	30.8	48.9	66.2	46.3
		CoInCo			
LexS	**14.0**	**29.7**	**38.0**	**59.2**	**50.5**
-w Pr.	12.9	26.5	37.6	58.5	47.8
-w Gl.	13.4	28.5	37.2	58.2	48.8
-w Sen.	13.6	29.9	37.2	58.3	49.2
-w Val.	12.7	27.0	35.9	57.4	46.6
Pr.	11.3	23.8	33.6	54.4	41.3
Gl.	8.4	16.7	29.6	47.2	33.6
Sen.	10.9	22.5	34.0	54.9	40.5
Val.	11.7	23.7	35.3	55.2	44.2

Table 3: Ablation study of LexSubCon: *Pr.* is the Proposal score using the mix-up embedding strategy. *Gl.* is the Gloss similarity score. *Sen.* is the Sentence Similarity score and *Val.* is the Validation score. *-w/o* indicates a LexSubCon framework **without** the specific feature.

In order to evaluate the effect of each feature on the performance of LexSubCon, we conducted an ablation study. The results are presented in Table 3. As Table 3 shows, LexSubCon achieved its best performance when it has access to information from all the features described in section 3. By testing the performance of the individual features, we observe that the gloss sentence similarity feature achieves the worst performance out of all the features. This is likely because many candidate words cannot be identified in Wordnet and thus we assign a zero value to their gloss sentence score. Another factor is that the models that were used for the selection of the most appropriate gloss for each word may introduce noise in the process of the gloss-similarity score model as they may select not-optimal glosses.

4.6 Candidate Ranking Task

We also evaluate LexSubCon in the candidate ranking task for both the LS07 an CoInCo dataset. In this sub-task the candidate substitution words are provided and the main task of the system is to create the most appropriate ranking of the candidates.

Method	LS07	CoInCo
LexSubCon	**60.6**	**58.0**
-w/o Pr.	58.8	56.3
-w/o Gl.	60.3	57.4
-w/o Sen.	59.8	57.1
-w/o Val.	56.8	53.8
Bert$_{s_p,s_u}$*	58.6	55.2
LexSubCon (trial+test)	60.3	58.0
Bert$_{s_p,s_u}$* (trial+test)	57.9	55.5
XLNet+embs	57.3	54.8
context2vec	56.0	47.9
Trans. learning	51.9	-
Sup. learning	55.0	-
PIC	52.4	48.3
Substitute vector	55.1	50.2
Addcos	52.9	48.3
Vect. space mod.	52.5	47.8

Table 4: Comparison of GAP scores (%) in previous published results in the candidate ranking task of our implementation of LexSubCon and Bert$_{s_p,s_u}$ (Zhou et al., 2019). We also provide the results on the entire dataset with (trial+test). Models: XLNet+embs (Arefyev et al., 2020), Context2vec (Melamud et al., 2016), Transfer learning (Hintz and Biemann, 2016), Supervised learning(Szarvas et al., 2013b), PIC (Roller and Erk, 2016), Substitute vector (Melamud et al., 2015a), Addcos (Melamud et al., 2015b), Vector space modeling (Kremer et al., 2014).

Table 4 provides the evaluation results in the candidate ranking task of LexSubCon and of the previous state-of-the art models. We report the results both on the test set and on the entire dataset (trial+test), in order to have a complete comparison as some of the previous state of the art models were evaluated on the entire datasets and some were evaluated only in the testing portion of the datasets. It can be observed that all the features have a positive effect on the performance of LexSubCon thus allowing it to outperform the previous state-of-the-art methods. Specifically, the results demonstrate the

Word	Sentence	Gold Ranking	LexSubCon	BERT$_{\text{based}}$
terrible	..have a **terrible** effect on the economy	awful, very bad, appalling, negative, formidable	horrible, horrific, awful	negative, major, positive
return	..has been allowed to **return** to its wild state	go back, revert, resume, regress	revert, retrovert, regress	recover, go, restore

Table 5: Examples of target words and their top lexical substitutes proposed by LexSubCon and BERT$_{\text{based}}$ model.

positive effect of the features on accurately ranking a list of potential candidates as LexSubCon outperforms the previous methods even in the scenario where they are all provided with the same substitution candidate list.

4.7 Qualitative Substitution Comparison

In Table 5, we provide different examples of target words and their top lexical substitutes proposed by LexSubCon and the BERT$_{\text{based}}$ model in order to demonstrate the effect of external lexical resources on the performance of a contextual model. As it can be observed, for the target word *terrible*, the BERT$_{\text{based}}$ model proposes a candidate word (*positive*) which may fit in the sentence but has the opposite meaning of the target word. However, LexSubCon provides semantically similar candidates by using information from different signals (e.g comparison of the definition of each word). In addition, for the target word *return*, our model identifies appropriate candidates that are not in the vocabulary of the contextual model (the word *regress*) by introducing candidates from a external lexical database. These examples showcase that enriching contextual models with external lexical knowledge can assist the model to provide more accurate candidates.

5 Extrinsic Evaluation: Data Augmentation

We evaluate the performance of LexSubCon in the context of textual data augmentation. Specifically, we conduct experiments on a popular benchmark text classification tasks of the English subjectivity/objectivity dataset (SUBJ) (Pang and Lee, 2004)[8]. The SUBJ dataset contains 5000 subjective and 5000 objective processed sentences (movie reviews). We train the LSTM model (with the same hyperparameters) which was used in (Wei and Zou, 2019) to measure the effect of different data augmentation techniques. We compare our method with previous state-of-the-art lexical substitution models and with other popular textual data augmentation techniques: (i) the back-translation technique (described in section 3.3) (ii) the EDA framework (Wei and Zou, 2019) which utilizes four operations of Synonym Replacement and Random Insertion/Swap/Deletion in order to create new text. Following the data generation algorithm in (Arefyev et al., 2020), LexSubCon creates new examples by sampling one word for each sentence, generating the appropriate substitute list for this word and sampling one substitute with probabilities corresponding to their substitute scores (which were normalized by dividing them by their sum) to replace the original word with the sampled substitute.

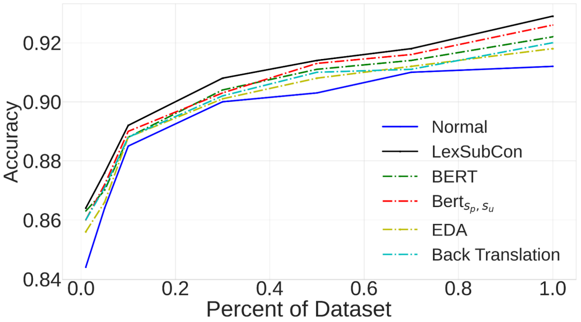

Figure 2: Accuracy with different train sizes for different text augmentation techniques on the SUBJ dataset.

Figure 2 demonstrates how data augmentation affects the classification depending on the size of the training set (Arefyev et al., 2020; Wei and Zou, 2019). As it is expected the effect of each text augmentation technique on the performance of the model becomes more significant while the size of the train set is reduced. Figure 2 also shows that the data created with lexical substitution have a more positive effect to the performance of the model than the other data augmentation techniques since back translation techniques may provide text that does not follow the syntactic rules of the target language and the EDA framework may create examples that could confuse the model by changing the structure of the sentence due the random insertion and

[8] license: https://tinyurl.com/t-license

swapping of words. Finally, since LexSubCon can create more accurate substitution candidates than the standard BERT model and the Bert$_{s_p,s_u}$ model, the texts that were created with LexSubCon have a more positive effect on the model's performance.

6 Conclusion

This paper presents LexSubCon, an end-to-end lexical substitution framework based on contextual embedding models. LexSubCon establishes a new mix-up embedding strategy that outperforms the previous SOTA strategy of word embedding dropout for the embedding of the target word for the task of predicting accurate candidate words. LexSubCon introduces the combined usage of features from contextual embedding models and external lexical knowledge bases in order to calculate accurately the semantic similarity between a target word and its candidates. We confirm that these features can improve the LexSubCon's performance as it outperforms other state-of-the-art results on two benchmark datasets.

As for future work, we plan to address the limitations of this study including: (i) examining the effect of using other models as the basis of our features (e.g. Albert (Lan et al., 2020)); (ii) exploring other candidate features for the ranking of the candidates (e.g. parser information (Szarvas et al., 2013a)) (iii) testing LexSubCon in datasets of other languages using multi-language lexical databases (e.g. MultiWordNet (Pianta et al., 2002) or BalkaNet (Oflazer et al., 2001)) to investigate further the model's general availability.

Ethical Consideration

Lexical substitution can be useful in various natural language processing (NLP) tasks such as textual data augmentation, paraphrase generation and text simplification. The results that we present in this paper suggest that contextual word embeddings models, such as our framework (LexSubCon), can be a valuable tool for providing accurate substitution candidates that can be further used in a variety of down-stream tasks.

We believe that there are many benefits of using our contextual embeddings models. For example, LexSubCon can be used as a data augmentation tool to provide artificial training data for tasks where the lack of sufficient training data may hurt the performance of the model. However, there are potential risks of over-relying on any lexical substitution tool. Particularly, a lexical substitution model can unintentionally change the meaning of the original text thus leading to erroneous conclusions.

References

Milam Aiken and Mina Park. 2010. The efficacy of round-trip translation for mt evaluation. *Translation Journal*, 14.

Nikolay Arefyev, Boris Sheludko, Alexander Podolskiy, and Alexander Panchenko. 2020. A comparative study of lexical substitution approaches based on neural language models.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations (ICLR)*.

Segun Taofeek Aroyehun and Alexander Gelbukh. 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Dzmitry Bahdanau, Tom Bosc, Stanislaw Jastrzebski, Edward Grefenstette, Pascal Vincent, and Yoshua Bengio. 2017. Learning to compute word embeddings on the fly. *ArXiv*, abs/1706.00286.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Katrin Erk and Sebastian Padó. 2010. Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 92–97, Uppsala, Sweden. Association for Computational Linguistics.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.

Aina Garí Soler, Anne Cocos, Marianna Apidianaki, and Chris Callison-Burch. 2019. A comparison of context-sensitive models for lexical substitution. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages

271–282, Gothenburg, Sweden. Association for Computational Linguistics.

Claudio Giuliano, Alfio Gliozzo, and Carlo Strapparava. 2007. FBK-irst: Lexical substitution task exploiting domain and syntagmatic coherence. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 145–148, Prague, Czech Republic. 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 Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 410–413, Prague, Czech Republic. 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)*, pages 118–129, Berlin, Germany. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3509–3514, Hong Kong, China. Association for Computational Linguistics.

Kazuaki Kishida. 2005. Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments. *NII Technical Reports*, 2005(14):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 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549, Gothenburg, Sweden. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. Sensebert: Driving some sense into bert. *ArXiv*, abs/1908.05646.

David Martinez, Su Nam Kim, and Timothy Baldwin. 2007. MELB-MKB: Lexical substitution system based on relatives in context. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 237–240, Prague, Czech Republic. Association for Computational Linguistics.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic. 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, Denver, Colorado. Association for Computational Linguistics.

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, Berlin, Germany. Association for Computational Linguistics.

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, Denver, Colorado. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Stamou Sofia Oflazer, Kemal Pala, and K. Christodoulakis. 2001. Balkanet: A multilingual semantic network for balkan languages.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 271–es, USA. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Emanuele Pianta, Luisa Bentivogli, and C. Girardi. 2002. Multiwordnet: Developing an aligned multilingual database.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. pages 3973–3983.

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, San Diego, California. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

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, Atlanta, Georgia. Association for Computational Linguistics.

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, Seattle, Washington, USA. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *ArXiv*, abs/1901.11196.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Deniz Yuret. 2007. KU: Word sense disambiguation by substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 207–214, Prague, Czech Republic. Association for Computational Linguistics.

Hongyi Zhang, M. Cissé, Yann Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. *ArXiv*, abs/1710.09412.

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. Association for Computational Linguistics.