LexSubDis: Knowledge-Integrated Lexical Substitution via Discriminative Ranking

Anonymous ACL submission

Abstract

Lexical substitution, a fundamental task in natural language processing, aims to replace target words with semantically equivalent or synonymous substitutes while preserving original sentence meaning. Although extensively explored, existing methods exhibit two major limitations: 1) inadequate investigation of embedding representations when target word contain subwords, and 2) excessive hyperparameters and computational complexity from multi-metric evaluation in candidate ranking. To address these issues, we propose LexSubDis, which constructs more MLM-compatible substitution mechanisms by averaging subword embeddings of target words and combining them with synonym embeddings. Moreover, we pioneer the introduction of discriminator models to assess semantic impacts of substitutions. Experimental results demonstrate that LexSubDis significantly reduces hyperparameters while achieving state-of-the-art performance under unsupervised learning on CoInCo dataset's *ootm* metric, offering novel insights and solutions for lexical substitution research.

1 Introduction

The lexical substitution task (McCarthy and Navigli, 2007) generates context-preserving candidate words for target terms, with applications spanning data augmentation (Morris et al., 2020), adversarial example generation (Jin et al., 2020), query optimization or rewriting (Jones et al., 2006), and word sense induction (Amrami and Goldberg, 2018).

The lexical substitution task comprises two phases: candidate generation and ranking, with core challenges in semantic preservation. Early approaches primarily leveraged pre-built resources (WordNet (Miller, 1995), PPDB (Ganitkevitch et al., 2013)) and statistical features (co-occurrence frequency, TF-IDF (Babych and Hartley, 2004)) for candidate extraction. Recent advancements driven by large language models have revolutionized this domain: Encoder-based architectures like BERT (Devlin et al., 2019) employ masked language modeling for context-aware substitution, while decoder-based models such as GPT (Radford et al., 2018) utilize autoregressive generation. Methodological innovations include a partial-mask dropout strategy (Zhou et al., 2019) that outperforms a conventional full-mask approach, and hybrid ranking mechanisms combining MLM/LM probabilities with embedding similarities (Arefyev et al., 2020). Comparative studies demonstrate XLNet (Yang et al., 2019) with embedding fusion achieves optimal performance through probability-based candidate sorting, surpassing context2vec (Melamud et al., 2016), ELMo (Peters et al., 2018), and RoBERTa (Liu et al., 2019). Notably, prompt-enhanced GPT-2 (Radford et al., 2019) training (Shi et al., 2024) significantly outperforms supervised baselines like GeneSis (Lacerra et al., 2021b) on the LS07 (McCarthy and Navigli, 2007) benchmark.

Regarding the input-output processing of target words, Arefyev et al. (2020) compared the MASK strategy with retaining original word (keep), finding the latter achieved better performance. Michalopoulos et al. (2022) employed mix-up technology, differing from traditional dropout (Zhou et al., 2019), by computing weighted averages between initial word embeddings of target words and their WordNet synonym embeddings, applying this operation only to the first subword when targets are split into multiple subwords. These methods inadequately explore the impact of subword-level segmentation of target words, and uniformly adopt the first subword representation at target positions as the holistic word representation. Additionally, existing approaches for candidate ranking focus on semantic changes in modified sentences, probabilistic similarity between candidates and target words, and comprehensive effects of substitutions on other words in the original sentence. These evaluation dimensions rely on combinations of multiple

weighted parameters, posing significant challenges for hyperparameter fine-tuning.

For the above two points, our work focuses on the choice of model processing methods on input and output and proposes a new candidate word evaluation method.

The main contributions of the paper were as follows:

- A new candidate rank method was proposed to improve the evaluation method. With fewer hyperparameters, this method achieves an improvement in the $ootm$ metric on the CoInCo dataset.

- The problem of the selection of the target word position of the MLM model was explored, including the selection of output and input.

2 Related Work

Lexical substitution aims to optimize the representation of text by using context-adapted word substitution while maintaining the same semantics. Current research focuses on two main directions: one is the construction and optimization of word substitution datasets, and the other is the innovative application of deep learning models in this task.

2.1 Lexical Substitution Resources

LST The Lexical Substitution Task (LST) dataset is SemEval 2007 Task 10 (McCarthy and Navigli, 2007). It encompasses 2,010 sentences with 201 target words, each appearing in 10 distinct contexts. The lexical coverage spans across nouns, verbs, adjectives, and adverbs, ensuring a diverse representation of grammatical categories. Annotation was conducted by five native English speakers, who collectively generated replacement terms for the target words.

CoInCo Concepts-In-Context (CoInCo) (Kremer et al., 2014) is a large-scale "all-words" lexical substitution resource designed to analyze word meaning in context. Unlike traditional "lexical sample" datasets (e.g., SemEval), which focus on isolated target words, this corpus annotates all content words in continuous text, providing a realistic distribution of lexical usage across contexts. It covers some 35K tokens of running text in which all 15.5K content words were labeled with at least 6 Synonyms using crowdsourcing methods. Annotators were able to see the whole sentence as well as two sentences of discourse context.

To address the data scarcity issue in lexical substitution tasks, the crowd-sourced TWSI dataset (Biemann, 2012) serves as a representative human-annotated resource, covering 1,012 high-frequency English nouns with 145,000 annotated sentences. For large-scale applications, AlaSca (Lacerra et al., 2021a) employed an automated pipeline for supervised data generation, while GENESIS (Lacerra et al., 2021b) leveraged a generative seq2seq model (Sutskever et al., 2014) to create contextualized examples, both demonstrating high validity in human evaluations.

2.2 Lexical Substitution Approaches

We will introduce the lexical substitution task from three perspectives: lexical generation models, lexical substitution methods, and lexical ranking.

Model Architecture The lexical substitution task is constrained by the scarcity of large-scale annotated data, limiting the application of supervised models. Existing approaches primarily fall into three categories (Lacerra et al., 2021a): knowledge-driven models leverage structured lexical resources like WordNet to extract synonyms; vector-space models compute candidate similarity through word embeddings (Melamud et al., 2016; Garí Soler et al., 2019; Peters et al., 2018); Transformer-based models (Vaswani et al., 2017), as an evolutionary extension of vector-space paradigms, employ pretrained architectures to generate deep contextual representations. Current research integrates these three paradigms, forming supervised and unsupervised methodological frameworks.

Substitution Methods The generation of lexical substitution can be divided into unsupervised and supervised. Unsupervised approaches do not require annotated data and mainly generate candidate words through pretrained models. For example, Zhou et al. (2019) proposed to incorporate BERT and a random masking mechanism in embeddings to enhance diversity. Supervised methods rely on annotated data to train models. Early studies (Szarvas et al., 2013a,b; Hintz and Biemann, 2016) transformed the lexical substitution task into a feature-based ranking problem through supervised learning frameworks. Subsequent work Qiang et al. (2023) designed a 6-layer Transformer encoder-decoder, outperforming pretrained model baselines,

and Shi et al. (2024) constructed a prompt framework based on GPT that significantly outperforms the generative baseline GeneSis (Lacerra et al., 2021b) on the LS07 dataset.

Ranking Mechanisms Candidate word ranking emphasizes overall sentence semantics and lexical correlations. Roller and Erk (2016) used exponential dot-product normalization to obtain probability values. Arefyev et al. (2020) ranked candidate words based on the predicted probabilities output by the model at the target position. Zhou et al. (2019) calculated the cosine similarity of token contextual representations before and after replacement, combining these with self-attention mechanism scores via weighted fusion to evaluate sentence-representation similarity after lexical replacement. Qiang et al. (2023) incorporated the text-generation evaluation metric BARTScore (Yuan et al., 2021), which assessed the semantic similarity of sentences with the replacement word embedded, replacing traditional word-embedding similarity methods.

3 LexSubDis Framework

We propose a lexical substitution task framework named LexSubDis (Figure 1), which consists of two parts: candidate generation and evaluation. In the candidate generation stage, we combine the Xlnet model (+embs) (Arefyev et al., 2020) with WordNet (Miller, 1995) to generate synonyms for the target word. In the evaluation stage, we rank the candidate words based on three types of scores. Notably, we introduce a discriminator model for the first time to score the candidates, aiming to assess their overall suitability.

3.1 Candidate Generator

In addition to generating candidate words based on model dictionaries, incorporating external resources can enhance performance. For example, Faruqui et al. (2015) integrated word vectors with information from semantic lexicons (WordNet) to enhance the semantic quality of word vectors. Michalopoulos et al. (2022) effectively improved candidate generation quality by integrating synonym sets of target words with multi-dimensional scoring metrics. Seneviratne et al. (2022) further incorporated WordNet-based definitions of target words and semantic similarity between sentences generated with substitute words into the candidate evaluation framework, providing more refined quantitative criteria for substitution effectiveness.

Different from the above work, We selected synonyms matching the target word's part-of-speech (POS), combining them with model-generated words through the following protocol: the top 10 probability-ranked words from the model's output were supplemented with up to 10 WordNet synonyms. When synonyms for the target word numbered fewer than 10, the deficit was filled by sequentially selecting from the model's next 10 highest-probability candidates. This methodology thereby ensured 20 candidate words per target lexical item.

3.2 Quality Discriminator

We employ the ELECTRA (Clark et al., 2020) model as a discriminator for candidate words. This model introduces novel pre-training tasks and frameworks by transforming the traditional generative Masked Language Model (MLM) pre-training task into a discriminative Replaced Token Detection (RTD) task, which focuses on determining whether the current token has been replaced by a language model. With fewer parameters and reduced data requirements, ELECTRA achieves comparable performance to the then SOTA model RoBERTa (Liu et al., 2019) while consuming only a quarter of its computational resources. Specifically, by replacing partial tokens in original sentences with contextually plausible alternatives, the model's objective becomes distinguishing replaced tokens from original ones.

3.3 Input-Output Strategies

Traditional word embedding models such as Word2Vec (Mikolov et al., 2013) fail to generate effective vector representations for out-of-vocabulary (OOV) words not present in the training data. FastText (Joulin et al., 2016) pioneered the decomposition of words into subword units and constructed word representations by summing subword vectors. Previous studies typically defaulted to applying strategies at target position, yet insufficient attention has been paid to cases where the target word is split into multiple subwords.

Input-Side Strategies In the context of input processing strategies, Michalopoulos et al. (2022) compared approaches including keep, mask, dropout, Gaussian noise, and mix-up, revealing that optimal model performance is achieved when the original embedding of the target word is combined with the

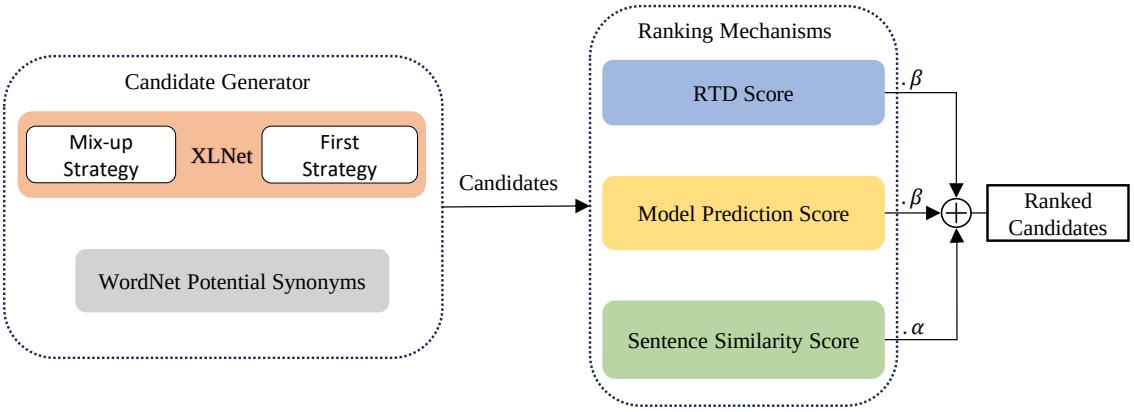

Figure 1: Schematic diagram of LexSubDis model architecture: Core components and WordNet-based potential synonym expansion module (optional).

average embedding of its synonyms. Specifically, the keep strategy (Arefyev et al., 2020) directly inputs the target word's original embedding into the model; the mask strategy replaces the entire target word with a mask token; the dropout strategy (Zhou et al., 2019) randomly zeros out specific dimensions of the target word's embedding vector with a predefined probability to improve generalization; the Gauss strategy injects Gaussian noise into the target word's embedding; and the mix-up strategy synthesizes new inputs by blending embeddings of the target word and its synonyms. Notably, these strategies are applied only to the first subword in the split subword sequence of the target word. To address this limitation, we propose a novel method: when a target word is segmented into multiple subwords, we unify its subword embeddings into a single-position representation, thereby mitigating the bias caused by processing only the initial subword.

Output-Side Strategies When a target word is split into multiple subwords, existing methods typically default to selecting the contextual representation of the first subword as the candidate representation, leveraging the bidirectional encoder architecture's inherent ability to capture contextual information. Building on this, we explore five subword fusion strategies: First (directly using the first subword's contextual representation), Pooling (Min/Max/Mean Pooling), Linear Weighting, and Exponential Weighting. The first subword representation remains conventional due to its computational simplicity. Pooling operations, originally from computer vision and later adapted to NLP (Bojanowski et al., 2017), aggregate subword contextual features: mean pooling averages all subword representations element-wise, max pooling selects dimension-wise maxima to emphasize salient features, and min pooling extracts minima to capture common characteristics. Linear and Exponential Weighting further model semantic distinctions between subwords. Both methods compute the target representation as (Eq. (1)):

$$h_{\text{target}} = \sum_{i=1}^{k} w_i \cdot h_i \quad \text{with} \quad \sum_{i=1}^{k} w_i = 1 \qquad (1)$$

where $h_i \in \mathbb{R}^d$ denotes the contextual representation of the i -th subword, k is the subword count, and weights w_i follow either linear or exponential allocation rules.

3.4 Lexical Substitutes Ranking

We propose three scores to evaluate candidate word quality. During assessment, it was observed that both the overall sentence semantics and inter-word interactions should be considered.

Model Prediction Score In pretrained language Models, when predicting the vocabulary at a target position, the token ID with the highest probability is selected as the prediction for that target position. This maximum posterior probability-based selection method ensures that the model always chooses the most likely token in the given context, thereby achieving semantic completion for the target positions. On this basis, Arefyev et al. (2019, 2020) formalized candidate modeling as C=(L,R), where T represents the target word, L and R denote left/right context respectively. The task aims to maximize the probability of substitute s given context C and target word T. According to Bayesian

rules, the formula is expressed as (Eq. (2)):

$$P(s \mid C, T) = \frac{P(C, T \mid s)\, P(s)}{P(C, T)} \tag{2}$$

In our experiments, we use $P(s \mid C, T)$ as the scoring function of the substitute model S_{p} for candidate words.

When constructing the candidate word set, we employ the top-10 words (Vijayakumar et al., 2016; Fan et al., 2018) with the highest predicted probabilities from the model as the base candidates, supplemented by 10 synonyms of the target word from WordNet. Since external resources lack probability scores generated by the model, we uniformly assign the fifth-highest probability value predicted by the model to these supplementary words. Empirical evidence shows that the top-5 strategy has been validated in multiple studies as an optimal choice for balancing prediction accuracy and computational efficiency. The fill-mask pipeline in Hugging Face Transformers returns the top-5 predictions by default[1].

Sentence Similarity Score Given an original sentence s containing a target word x_i, we generate an updated sentence s' by replacing the target word with a candidate substitute. The updated sentence can be represented as:

$$s' = (x_1, \ldots, x_i', \ldots)$$

For each candidate substitute, we compute a sentence-level semantic similarity score between the original sentence s and the updated sentence s' using Sentence-BERT (Reimers and Gurevych, 2019). Specifically, we obtain the sentence embeddings $\text{SBERT}(s)$ and $\text{SBERT}(s')$, and calculate their cosine similarity as follows:

$$S_{\text{s}} = \cos(\text{SBERT}(s), \text{SBERT}(s'))$$

This score measures the degree to which the semantic meaning of the sentence is preserved after the substitution.

RTD Score In the Electra model (Clark et al., 2020), the output of the Discriminator is the unnormalized logits, which are converted into the probability that each token is a replaced word by applying the sigmoid function. Specifically, let the logit for the i-th token in the original sentence be z_i. After applying the sigmoid function, we obtain the corresponding probability:

$$p_i = \sigma(z_i)$$

Similarly, for the sentence after candidate word replacement, the probability for the i-th token is given by:

$$p_i^* = \sigma(z_i^*)$$

where $\sigma(\cdot)$ denotes the sigmoid function. To evaluate the impact of replacing the target word with a candidate word on the overall sentence, we compute the average of the absolute differences between the probabilities of corresponding token positions in the original and modified sentences. The metric, denoted as Sr, is defined as:

$$S_{\text{r}} = \frac{1}{N} \sum_{i=1}^{N} |p_i - p_i^*| \tag{3}$$

where N represents the total number of tokens in the sentence. This metric is analogous to the Mean Absolute Error (MAE) and quantifies the overall effect of the replacement on the model's output.

In summary, the calculation formula for the candidate words is shown in (Eq. (4)) :

$$S_{\text{srp}} = \alpha \cdot S_s + \beta \cdot ((1 - S_r) + S_p) \tag{4}$$

where α and β are hyperparameters. Noted that S_r represents the token substitution error between the original and new sentences, with a value range of $[0, 1]$, where smaller values indicate lower error. To intuitively quantify the scores of candidate words, we perform a transformation by taking the difference from 1 (i.e., $1 - S_r$).

4 Experiments

4.1 Datasets In Experiments

To better compare with prior work, we adopt two highly representative lexical substitution datasets: the SemEval 2007 dataset (McCarthy and Navigli, 2007) (abbreviated as LS07) and the CoInCo dataset (Kremer et al., 2014) (abbreviated as LS14). Both datasets contain original sentences, original word indices, original word lemmas, gold substitute words, gold substitute weights, and original word parts of speech, where the parts of speech cover adverbs, nouns, verbs, and adjectives. Notably, gold substitutes may include multi-word phrases. As LS07 contains 8 entries missing gold substitutes, these were filtered out. The weights assigned

[1] https://huggingface.co/docs/transformers

Method	best	bestm	oot	ootm	P@1	P@3
				LS07		
Bert with dropout (Zhou et al., 2019)	20.30	34.20	55.40	68.40	51.10	-
XLNet+embs (Arefyev et al., 2020)	21.32	37.80	55.04	73.90	50.56	36.29
LexSubCon (Michalopoulos et al., 2022)	21.10	35.50	51.30	68.60	51.70	-
CILex3 (Seneviratne et al., 2022)	**23.31**	**40.98**	**56.32**	**74.88**	**55.96**	**38.50**
LexSubDisc*	20.46	35.85	55.34	74.12	48.23	34.65
LexSubDisc	20.55	35.99	55.97	73.84	48.48	36.21
				CoInCo		
Bert with dropout	14.50	33.90	45.90	69.90	56.30	-
XLNet+embs	15.09	33.02	45.06	71.85	52.57	39.67
LexSubCon	14.00	29.70	38.00	59.20	50.50	-
CILex3	**16.39**	**35.80**	**46.87**	72.98	**57.25**	**42.49**
LexSubDisc*	14.24	32.68	44.75	73.09	51.37	38.84
LexSubDisc	14.35	33.01	46.50	**74.66**	51.75	41.20

Table 1: Results of the best implementation of our approach and previous unsupervised models for the LS07 and CoInCo datasets. Note: * indicates the incorporation of WordNet synonym expansion during candidate generation. Best values are bolded.

to each gold substitute reflect their selection frequency by annotators. In preliminary studies, to compute the Generalised Average Precision (GAP) metric (Kishida, 2005), the candidate substitution sets were constructed based on WordNet in previous work (Roller and Erk, 2016; Szarvas et al., 2013a), which included not only synonyms from target word synsets but also semantically similar words and words with entailment relationships.

4.2 Experimental Setup

We adopted the measurement metrics proposed in (McCarthy and Navigli, 2007), including $best$, $bestm$, out of ten (oot), and $ootm$. The best metric validates the optimal accuracy of model-generated substitutes, while the oot metric evaluates the coverage degree of candidate substitutes against gold substitutes. The suffix "m" denotes mode: if no unique maximum-weight substitute (i.e., no mode) exists in the gold substitutes where one weight significantly exceeds others, the corresponding data entry is excluded from the statistical calculation[2]. Note that the $best$ and oot metrics are used to evaluate the top-1 and top-10 prediction performance of the model (Shi et al., 2024). To contrast with (Arefyev et al., 2020; Seneviratne et al., 2022), we computed the precision rates for the top-1 and top-3 predictions (P@1 and P@3), and further calculated the recall rate for the top-10 predictions (R@10).

[2]The scoring measures are as described in the document at http://nlp.cs.swarthmore.edu/semeval/tasks/task10/task10documentation.pdf.

Given the demonstrated versatility of BERT model in short-text processing tasks, the BERT-Large, Cased model was designated as the baseline for comparative analysis of input-output strategies. Building upon the methodology established in (Arefyev et al., 2020) with augmentation from the WordNet lexical database, candidate substitutions were generated through XLNet-Large, Cased model. These modified sentences were subsequently processed by the ELECTRA-Large-Discriminator to capture errors. Inter-sentence semantic divergence was quantified using the all-roberta-large-v1 variant of SentenceBERT (Reimers and Gurevych, 2019). Out-of-vocabulary tokens were handled via subword embedding averaging, with unmapped subword units assigned zero-valued vectors.

In this study, the batch size was set to 50, with each data point selecting the top 20 words by probability, and each target word corresponding to up to 10 synonyms. For the input strategy, we adopted a dropout rate of 0.3 consistent with (Zhou et al., 2019), set the standard deviation of Gaussian noise to 0.01, and configured the mix-up parameter as 0.25 following (Michalopoulos et al., 2022). Additionally, the initial input vector incorporates embeddings of up to 10 synonyms only. In terms of the output strategy, a linear weighting method was used to gradually decrease weights from 1.0 to 0.5, alongside an exponential weighting method with an exponential decay rate of 0.9. For candidate word score calculation, α and β were set to 0.7 and

0.3 respectively, both within the range $[0, 1]$ and summing to 1.

4.3 Model Comparison

This study primarily compares recent work on lexical substitution tasks using unsupervised learning. Our experiments are built upon the foundation of (Arefyev et al., 2020), similarly implemented on the XLNet framework while incorporating target word embeddings and their approximate embeddings in the model (+embs). For target word input processing, we adopted five strategies from (Michalopoulos et al., 2022). Instead of simply replacing the first subword embedding of target words, we first averaged the subwords of target words before applying these five strategies. We conducted experimental comparisons between these two processing approaches (see Section 4.4). Regarding the incorporation of WordNet synonyms, Michalopoulos et al. (2022) selected 30 synonyms, whereas we chose only 10 as supplementary to avoid introducing excessive noise.

After completing the experimental setup, we analyzed and compared the results (see Table 1). Our model achieved optimal performance on the *ootm* metric of the CoInCo dataset. Experiments revealed that on the CoInCo dataset with larger data volume and target words containing more subwords, LexSubDis outperformed LexSubCon across all metrics and surpassed XLNet+embs in top-3 and top-10 candidate metrics. However, for the top-1 metric (i.e., determining whether the highest-scoring candidate appears in gold-standard substitutions), this method underperformed compared to direct model-generated candidate ranking approaches. We conducted ablation experiments to investigate this (see Table 4). Additionally, incorporating WordNet synonyms of target words affected model performance across metrics. Data shows that synonym incorporation only slightly outperformed non-synonym scenarios on the ootm metric of LS07.

4.4 Input-Side Strategies Evaluation

We evaluate five strategies based on the BERT model using the CoInCo dataset. For target word processing, we test two approaches: one exclusively applied to the first subword position of the target token (Michalopoulos et al., 2022), and the other operating on the averaged embeddings of subword units. Experimental results (see Table 2) demonstrate that the mix-up strategy incorporating synonym representations achieves optimal performance on the P@1 and P@3 metric. Furthermore, the subword-averaged processing approach consistently outperforms the first-position-only method across all strategies.

Stra.	bestm	oot	ootm	P@1	P@3
Mix.	**25.1**	36.5	60.0	**43.8**	**32.9**
	24.5	36.0	59.2	42.9	32.2
Keep	24.5	**36.7**	**61.3**	43.1	32.7
	24.3	36.4	60.8	42.6	32.4
Drop.	24.1	36.4	60.8	42.5	32.3
	23.9	36.0	60.1	42.0	30.0
Gaus.	24.5	**36.7**	61.2	43.1	32.7
	24.3	36.4	60.8	42.6	32.4
Mask	14.2	26.0	43.1	26.7	20.8
	13.7	25.1	41.5	25.6	19.9

Table 2: Comparison of different subword-to-word strategies on Lexical Substitution performance. Each strategy is evaluated in two ways: the first row applies average pooling over subwords, and the second uses the first subword only. Best values are bolded.

4.5 Output-Side Strategies Evaluation

Previous lexical substitution research did not incorporate subword information, always assuming that the context representation of a target word was given by the position of its first subword. We experimented with six subword context representation strategies (see Table 3 for details). To compare with (Arefyev et al., 2020), we evaluated them using precision, recall, and GAP (Generalized Average Precision) metrics. The results indicate that the performance differences across the strategies are not significant, but methods that comprehensively utilize subword information (such as weighted averaging of subword vectors) show a slight improvement in overall performance compared to the baseline method that only uses the first subword.

4.6 Ablation Study

To thoroughly evaluate the impact of synonyms and the three scoring metrics, this study employs the XLNet (+embs) model (Yang et al., 2019; Arefyev et al., 2020) to generate 20 candidate words, which are then combined with 10 synonyms extracted from WordNet. The combination of candidate words and synonyms follows the method described in Section 3.1, with experimental results presented in Table 4. Based on this experimental framework, our findings reveal that: The introduction of synonyms leads to decreased accuracy in can-

Stra.	P@1	P@3	R@10	GAP
LS07				
1st.	38.04	**27.75**	**39.62**	54.44
Min	37.79	27.55	39.29	54.35
Max	37.94	27.71	39.56	54.44
Mean	37.97	27.66	39.48	54.42
Lin.	**38.12**	**27.75**	39.58	**54.46**
Exp.	38.09	27.73	39.60	**54.46**
CoInCo				
1st.	43.02	32.91	29.01	50.64
Min.	43.04	32.66	28.79	50.42
Max	43.33	33.25	29.31	50.64
Mean	**43.61**	**33.31**	**29.37**	**50.67**
Lin.	43.02	32.91	29.09	50.64
Exp.	42.78	32.70	28.91	50.59

Table 3: Performance of the six subword context representation strategies on the LS07 and CoInCo datasets. Best values are bolded.

Meth.	best	bestm	oot	ootm	P@1
LS07					
S_p	20.1	35.3	54.6	72.3	47.3
S_p^*	20.0	35.2	49.5	68.1	47.1
S_s	13.6	22.4	51.6	68.6	33.0
S_s^*	11.5	18.5	52.8	70.8	27.6
S_r	12.7	19.7	51.5	67.3	33.6
S_r^*	10.4	15.9	49.7	66.2	27.2
S_{sr}	16.8	28.0	53.9	70.9	41.3
S_{sr}^*	14.0	23.7	54.2	72.5	33.7
S_{srp}	**20.6**	**36.0**	**56.0**	73.8	**48.5**
S_{srp}^*	20.5	35.9	55.3	**74.1**	48.2
CoInCo					
S_p	14.0	32.2	45.4	73.4	50.7
S_p^*	14.0	32.0	40.7	68.7	50.6
S_s	9.7	20.2	43.4	69.7	37.5
S_s^*	7.2	14.7	43.0	69.9	28.5
S_r	8.3	15.8	40.5	63.5	35.6
S_r^*	6.0	11.6	38.1	61.7	26.1
S_{sr}	11.7	24.8	44.7	71.0	45.2
S_{sr}^*	8.5	18.0	43.6	70.5	33.4
S_{srp}	**14.4**	**33.0**	**46.5**	**74.7**	**51.8**
S_{srp}^*	14.2	32.7	44.8	73.1	51.4

Table 4: Note: * indicates the incorporation of WordNet synonym expansion during candidate generation. S_p, S_r, and S_s denote the Model Prediction Score, RTD Score, and Sentence Similarity Score, respectively; S_{srp} represents the combined score integrating all three metrics. Best values are bolded.

didate ranking under both top-1 and top-3 metrics. S_p plays a crucial role in candidate ranking as it effectively integrates contextual information of target words, while S_r and S_s respectively focus on assessing the global impacts of substituted words on semantic integrity and syntactic structure. Experimental results show that the synergistic combination of these three scoring mechanisms yields optimal candidate ranking performance.

5 Conclusion

This study first conducts an in-depth investigation into the input and output processing mechanisms for target words. On the input side, by integrating the subword embedding representations of the target word, the method not only facilitates the incorporation of synonym information but also effectively reduces the computational complexity during candidate word evaluation. On the output side, averaging the contextual embeddings of the subwords corresponding to the target word enables the acquisition of more robust semantic representations. Moreover, this study is the first to apply a discriminator model trained through the Replaced Token Detection (RTD) task to candidate word ranking. Compared to existing approaches, this method offers lower computational costs and better adaptability to the lexical substitution task.

Limitations

In this study, we have not explored methods for generating synonym phrases, but have only combined external resources with the model's vocabulary for candidate word generation. Meanwhile, we solely employed unsupervised learning methods without implementing specific fine-tuning optimizations for the lexical substitution task. For the model's $top-k$ sampling strategy, we simply fix the value of k without dynamically adjusting it based on the probability differences among the candidate tokens. As limitations of the current work, we plan to further investigate the application potential of autoregressive models in lexical substitution tasks in subsequent studies, and attempt to integrate multiple natural language processing models and technical approaches to address related issues more systematically.

References

Asaf Amrami and Yoav Goldberg. 2018. Word sense induction with neural bilm and symmetric patterns.

arXiv preprint arXiv:1808.08518.

Nikolay Arefyev, Boris Sheludko, and Alexander Panchenko. 2019. Combining lexical substitutes in neural word sense induction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 62–70, Varna, Bulgaria. INCOMA Ltd.

Nikolay Arefyev, Boris Sheludko, Alexander Podolskiy, and Alexander Panchenko. 2020. Always keep your target in mind: Studying semantics and improving performance of neural lexical substitution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1242–1255, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bogdan Babych and Tony Hartley. 2004. Extending the bleu mt evaluation method with frequency weightings. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 621–628.

Chris Biemann. 2012. Turk bootstrap word sense inventory 2.0: A large-scale resource for lexical substitution. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 4038–4042, Istanbul, Turkey. European Language Resources Association (ELRA).

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *Preprint*, arXiv:2003.10555.

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.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. 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.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 758–764.

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.

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.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*, pages 387–396.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Kazuaki Kishida. 2005. *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. National Institute of Informatics Tokyo, Japan.

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.

Caterina Lacerra, Tommaso Pasini, Rocco Tripodi, Roberto Navigli, and 1 others. 2021a. Alasca: an automated approach for large-scale lexical substitution. In *IJCAI*, pages 3836–3842.

Caterina Lacerra, Rocco Tripodi, and Roberto Navigli. 2021b. GeneSis: A Generative Approach to Substitutes in Context. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10810–10823, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis,

Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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, 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.

George Michalopoulos, Ian McKillop, Alexander Wong, and Helen Chen. 2022. LexSubCon: Integrating knowledge from lexical resources into contextual embeddings for lexical substitution. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1226–1236, Dublin, Ireland. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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

John X Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*.

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 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.

Jipeng Qiang, Kang Liu, Yun Li, Yunhao Yuan, and Yi Zhu. 2023. ParaLS: Lexical substitution via pretrained paraphraser. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3731–3746, Toronto, Canada. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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.

Sandaru Seneviratne, Elena Daskalaki, Artem Lenskiy, and Hanna Suominen. 2022. CILex: An investigation of context information for lexical substitution methods. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4124–4135, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Ning Shi, Bradley Hauer, and Grzegorz Kondrak. 2024. Lexical substitution as causal language modeling. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 120–132, Mexico City, Mexico. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

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.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: evaluating generated text as text generation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

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.