# ParaLS: Lexical Substitution via Pretrained Paraphraser

**Jipeng Qiang** and **Kang Liu** and **Yun Li** * and **Yunhao Yuan** and **Yi Zhu** *
School of Information Engineering, Yangzhou University, Yangzhou, China
{jpqiang, liyun, yhyuan, zhuyi}@yzu.edu.cn, liukang@stu.yzu.edu.cn

## Abstract

Lexical substitution (LS) aims at finding appropriate substitutes for a target word in a sentence. Recently, LS methods based on pretrained language models have made remarkable progress, generating potential substitutes for a target word through analysis of its contextual surroundings. However, these methods tend to overlook the preservation of the sentence's meaning when generating the substitutes. This study explores how to generate the substitute candidates from a paraphraser, as the generated paraphrases from a paraphraser contain variations in word choice and preserve the sentence's meaning. Since we cannot directly generate the substitutes via commonly used decoding strategies, we propose two simple decoding strategies that focus on the variations of the target word during decoding. Experimental results show that our methods outperform state-of-the-art LS methods based on pre-trained language models on three benchmarks.

## 1 Introduction

Lexical substitution (LS) in context (Hintz and Biemann, 2016; Zhou et al., 2019; Arefyev et al., 2020) is an extremely powerful technology that can be used as a backbone of various NLP applications such as writing assistance (Lee et al., 2021), word sense disambiguation (McCarthy, 2002), and lexical simplification (Paetzold and Specia, 2016; Qiang et al., 2021a,b). Compared with traditional LS methods based on linguistic databases (e.g., WordNet) (Hassan et al., 2007; Yuret, 2007) or word embedding models (Melamud et al., 2015a,b), LS methods based on pretrained language models have made remarkable progress in generating substitutes by considering the context (Zhou et al., 2019; Qiang et al., 2021a; Michalopoulos et al., 2022; Seneviratne et al., 2022). These methods feed the sentence into BERT (Devlin et al., 2018) or XLNet (Yang et al., 2019) to obtain the top probability words corresponding to the target word as the substitute candidates. However, they have the following two limitations.

(1) The predictability of words is greatly influenced by the surrounding context, with little regard for preserving the sentence's meaning. As illustrated in Table 1, the utilization of pretrained models often leads to the generation of ill-suited words, such as "wet", "flat" and "cold", due to their contextual relevance and similarity to the target word.

(2) The utilization of subword techniques in pretrained models precludes the selection of multi-token words as substitutes, as they only generate the most probable single tokens. For instance, the words "desiccated" and "facilitated" would not be offered as a substitution for the target word "dry" as seen in Table 1.

To address the limitations mentioned above, we study how to generate substitutes via paraphrase modeling. Recent neural paraphrasers based on encoder-decoder framework (Wieting and Gimpel, 2017; Hu et al., 2019) produce fluent, meaning-preserving English paraphrases but contain variations in word choice. Therefore, our idea is whether we can decode the substitute candidates from the hidden representation of the target word. In this way, the substitutes are not only semantically consistent with the target word and fit in the context, but also can preserve the sentence's meaning. The meaning-preserving properties of a paraphraser can aid in addressing the first limitation, while autoregressive paraphrasers can address the second limitation. To the best of our knowledge, paraphraser for LS task has not yet been explored, as current decoding methods focus on lexical variations within the entire sentence rather than the target word, resulting in a scarcity of appropriate substitutes for the target word.

To specifically focus on lexical variations of the

*Corresponding author.



*Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*
*Volume 1: Long Papers*, pages 3731–3746

| | |
|---|---|
| **Sent1** | surprisingly in such a **dry** continent as Australia , salt becomes a · · · |
| **Labels** | arid, waterless |
| **BERT** | wet,arid,moist,humid,damp |
| **XLNet** | wet, flat, moist, desert, cold |
| **Ours** | desiccated,drought,arid,dead, parched |
| **Sent2** | remember that the delegates ' life is not always **easy**. |
| **Labels** | simple, trouble free, undemanding, uncomplicated, straightforward |
| **BERT** | simple, hard,complicated, difficult, exciting |
| **XLNet** | cheap, simple, quick, hard, fast |
| **Ours** | simple, light, good, ease, facilitated |

Table 1: The top five substitutes of two instances in LS07 dataset using BERT (Zhou et al., 2019), XLNet (Seneviratne et al., 2022) and our method. The target word of each sentence is bolded, and the suitable substitutes are marked in red.

target word during the decoding process, we propose two new decoding strategies. (1) Our first strategy, referred to as ParaLS, proposes fixing prefixes of the target word. This approach initiates the decoding process by mandating that the decoder begins with the target word's prefixes in the sentence, to subsequently generate the probability distribution of the target word's position. The words with the highest probabilities are then fixed and used when decoding the remaining words, with selected words of the target word's position in the paraphrases being selected as substitute candidates. (2) The second strategy, referred to as ParaLS$\star$, is proposed to address the oversight of suffixes in the first strategy. Inspired by NEUROLOGIC A$\star$esque (Lu et al., 2022), which incorporates heuristic estimates of future cost, we adapt it to estimate of the words in suffixes.

To the best of our knowledge, ParaLS is the first LS method that can produce substitute candidates by considering preserving the sentence's meaning. On three benchmarks, ParaLS and ParaLS$\star$ achieve state-of-the-art performance across various evaluation metrics. Moreover, ParaLS$\star$ without the step of substitute ranking outperforms all existing methods with the step of substitute ranking.

Additionally, we propose a novel strategy for the step of substitute ranking by text generation evaluation metrics BARTScore (Yuan et al., 2021) and BLEURT (Sellam et al., 2020). Our method embeds each substitute into the original sentence to create an updated version. By using BARTScore and BLEURT to compute the relationship between the original and updated sentences, they can quantify the extent to which the meaning of the original sentence has been preserved by each substitute. Experimental results show that substitute ranking using only BARTScore outperforms the previous state-of-the-art ranking methods when the same substitution candidate lists are provided for two popular LS benchmarks. The code and the experimental results are source-opened in Github [1].

## 2 Related Work

**Lexical Substitution.** LS methods generally consist of two steps: substitute generation and substitute ranking. Previous LS methods utilize linguistic databases (e.g., WordNet) (Hassan et al., 2007; Yuret, 2007) or word embedding models (Melamud et al., 2015b,a; Qiang and Wu, 2021) to extract synonyms or highly similar words for a target word, and then sort them based on their appropriateness in context. These methods overlook the context of the target word while generating substitute candidates, thereby inevitably generating a plethora of irrelevant candidates that may impede the subsequent ranking phase.

Recent LS methods based on pretrained language models have attracted much attention (Zhou et al., 2019; Lacerra et al., 2021a; Michalopoulos et al., 2022), in which the pretrained BERT is the most widely used one. Zhou et al. (Zhou et al., 2019) apply dropout to the embeddings of target words, Michalopoulos et al. (Michalopoulos et al., 2022) propose a new mix-up embedding strategy by incorporating the knowledge of WordNet into the prediction process of BERT, and Lin et al. (Lin et al., 2022) proposed an auxiliary gloss regularizer module to BERT pre-training. Lacerra et al. (Lacerra et al., 2021b,a) tried to train pretrained language models by merging the development set of two LS datasets (CoInCo and TWSI). The current work (Arefyev et al., 2020; Seneviratne et al., 2022) sought to evaluate all existing pretrained language models, and found that combining the prediction of pretrained language models XLNet and Word2Vec achieved the best results.

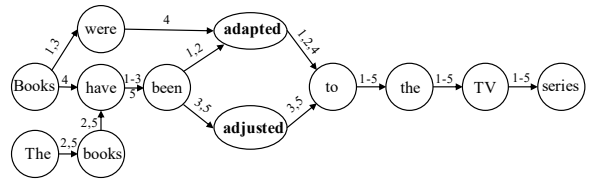
Overall, pretrained language modeling-based LS methods consider contextual information of tar-

[1] https://github.com/qiang2100/ParaLS

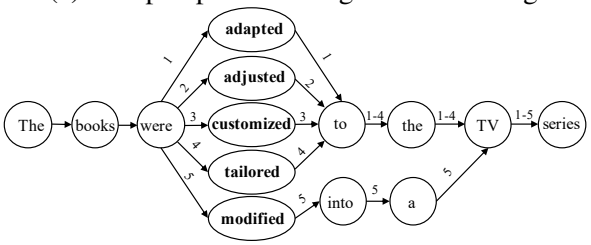
get words when generating substitute candidates, but do not concern with the impact of applying substitutes on sentence meaning. In contrast to the aforementioned methods, we try to utilize the knowledge of a pretrained paraphraser to generate substitute candidates.

**Lexical Substitution using Paraphrases.** A few studies (Pavlick and Callison-Burch, 2016; Kriz et al., 2018) find substitute candidates for complex words from a large-scale paraphrase rule database, e.g., PPDB (Ganitkevitch et al., 2013) or its variations (Pavlick et al., 2015; Pavlick and Callison-Burch, 2016). A paraphrase rule database consists of large-scale lexical paraphrase rules (e.g., "berries→strawberries") that are extracted from large-scale paraphrase sentence pairs, such as ParaNMT (Wieting and Gimpel, 2017) or ParaBank (Hu et al., 2019). These methods do not take into account the context as linguistic resource-based LS methods do. In this paper, we generate substitute candidates of target words using the pretrained paraphrase model instead of paraphrase rule databases or paraphrase databases.

**Decoding Strategies.** Paraphrase generation can be regarded as a monolingual machine translation task that transforms expressions of an input sentence while retaining its meaning (Wieting and Gimpel, 2017). Neural paraphrasers primarily rely on the encoder-decoder framework, achieving inspiring performance gains over the traditional approaches (Lu et al., 2022). Beam search decoding is the most common method for inference, which decodes the top-$K$ sequences in a greedy left-to-right fashion. When $K$ is set to 1, beam search decoding is changed into greedy search decoding. In recent years, beam search decoding has had multiple variants to deal with various task-specific and diversity/fluency trade-off of outputs, such as noise beam decoding (Cho, 2016), iterative beam decoding (Kulikov et al., 2019), clustered beam decoding (Tam, 2020) and diverse beam decoding (Vijayakumar et al., 2018). To enable constrained generation, NEUROLOGIC A⋆esque (Lu et al., 2022) explicitly decodes future text to estimate the viability of different paths for satisfying constraints. In contrast to the above decoding strategies, our decoding strategies focus solely on enhancing the diversity of the target word's variation.




(a) Five paraphrases using beam decoding.




(b) Five paraphrases using our decoding strategy via fixing prefixes.

Figure 1: The paraphrases of the sentence "The books were adapted into a television series" using two different decoding methods. "adapted" is the hypothetical target word. Figure (a) shows the normal paraphrases of beam decoding with beam size 5, and Figure (b) shows the first 5 paraphrases of our decoding method by forcing the decoder to begin with prefixes "The books were" of the target word. We have easily access to substitute candidates of the target word "adapted" from the paraphrases using our decoding method.

## 3 Method

Given a given sentence $\mathbf{x} = \{x_1, x_2, ..., x_t, ..., x_n\}$ and the target word $x_t$, we need a pretrained paraphraser based on an autoregressive model, instead of a pretrained language modeling like existing LS methods (Zhou et al., 2019; Michalopoulos et al., 2022; Seneviratne et al., 2022), e.g., BERT or XLNet. LS method consists of two steps: substitute generation and substitute ranking. After feeding sentence $\mathbf{x}$ into the paraphraser, we aim to extract substitute candidates for the target word $x_t$ by two novel decoding strategies (Section 2.2). Then, we rank the candidates to choose the most appropriate substitution without modifying the meaning of $\mathbf{x}$ (Section 2.3).

### 3.1 Motivation

Recent neural paraphrasers primarily rely on the encoder-decoder learning framework on a large-scale paraphrase dataset, achieving inspiring performance gains over traditional methods (Meng et al., 2021; Kadotani et al., 2021). Many languages including English, French, German, Chinese, and Spanish own large-scale paraphrase datasets. For example, in English, ParaBank2 (Hu et al., 2019)

consists of 19,370,798 sentence pairs.

Given an input sentence $\mathbf{x}$ and its corresponding paraphrase $\mathbf{y}$, we consider standard left-to-right, autoregressive models, $p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{|\mathbf{y}|} p_\theta(y_t|\mathbf{y}_{<t}, \mathbf{x})$, and omit $\mathbf{x}$ to reduce clutter. Decoding consists of solving,

$$\mathbf{y}_* = \arg\max_{Y \subseteq \mathcal{Y}} F(\mathbf{y}) \tag{1}$$

where $\mathcal{Y}$ is the set of all sequences, and the objective $F(\mathbf{y})$ is $\log p_\theta(\mathbf{y})$.

If we want to generate multiple paraphrases of sentence $\mathbf{x}$, the beam search decoding is widely used by the auto-regressive method, which maintains a beam of $K$ possible generations, updating them incrementally by ranking their extensions via the model likelihood. Since beam search decoding aims to find the most-probable hypothesis for the whole sentence during decoding, it is difficult to extract multiple substitute candidates for the target word from the generated paraphrases, as shown in Figure 1(a).

Since beam decoding concerns lexical variations of the whole sentence instead of the target word, there are no sufficient appropriate substitutions that can be discovered for the target word if we directly extract the substitute candidates from the paraphrases using the beam decoding. We will propose two novel decoding strategies, ParaLS and ParaLS$\star$, for the paraphraser that are specifically engineered to harness lexical variations of the target word.

### 3.2 Substitute Generation

Substitute generation aims to generate substitute candidates for the target word based on its context. We will generate the candidates during the process of decoding.

The process of decoding method can be treated as a discrete search, in which $states$ are partial prefixes, $\mathbf{y}_{<t}$, $actions$ are tokens in vocabulary $\mathcal{V}$ (i.e., $y_t \in \mathcal{V}$), and $transitions$ add a token to prefixes, $\mathbf{y}_{<t} \circ y_t$. Each step of decoding consists of (1) expanding a set of candidate next-states, (2) scoring each candidate, and (3) selecting the best $K$ candidates.

(1) **ParaLS: Decoding by Fixing Prefixes.** Given a sentence $\mathbf{x}$ and a target word $x_t$, we force the decoder to begin with prefixes $\mathbf{x}_{<t}$ of the target word, and decode succeeding token $y_t$ to estimate the probability distribution of the vocabulary $p(y_t|\mathbf{x}_{<t})$. We select the top $K$ tokens $Y_t$ with the highest probability in the distribution as the results of decoding.

$$Y_t^{'} = \{\mathbf{y}_{<t} \circ y_t | \mathbf{y}_{<t} = \mathbf{x}_{<t}, y_t \in \mathcal{V}\}$$

$$Y_t = \underset{(\mathbf{y}_{<t} \circ y_t) \in Y_t^{'}}{\arg topK} \{f(\mathbf{y}_{<t}, y_t)\} \tag{2}$$

where $f(\cdot)$ is scoring function that approximates the objective $F$.

The decoding phase by fixing prefixes $\mathbf{x}_{<t}$ is crucial to generate substitute candidates since we forcibly generate $K$ different tokens $Y_t$ with the highest probability. In this case, these generated tokens are not only semantically consistent with the target word and fit in the context, but also preserve the sentence's meaning. Since one word may comprise two or more tokens, we adopt greedy search decoding to select the token that has the maximum probability for each preceding token, until reaching the end symbol "EOS" of one sentence. These $K$ words are considered as substitute candidates, after eliminating the morphological derivations of the target word. As depicted in Figure 1(b), our decoding strategy concentrates on lexical variations of the target word.

(2) **ParaLS$\star$:Decoding with Lookahead Heuristics.** ParaLS, by fixing prefixes, takes into account only prefixes $\mathbf{x}_{<t}$ without accounting for suffixes $\mathbf{x}_{>t}$. In this manner, the top $K$ tokens $Y_t$ by Equation (2) may be one word of suffixes. Drawing inspiration from the A$^*$ search algorithm (Hart et al., 1968) and NEUROLOGIC A$^\star$esque (Lu et al., 2022), ParaLS$\star$ will incorporate an estimate of the words of suffixes into the prediction of $p(y_t|\mathbf{x}_{<t})$, replacing Equation 2 with:

$$Y_t = \underset{(\mathbf{y}_{<t} \circ y_t) \in Y_t^{'}}{\arg topK} \{maxF(\mathbf{y}_{<t}, y_t, \mathbf{y}_{>t}\} \tag{3}$$

where $\mathbf{x}_{>t}$ represents suffixes, $\mathbf{y}_{<t}$ equals $\mathbf{x}_{<t}$, and $\mathbf{y}_{>t}$ equals $\mathbf{x}_{>t}$.

ParaLS$\star$ enhances the ParaLS scoring function by incorporating an estimate of suffixes satisfaction. Our key addition is a lookahead heuristic that adjusts a candidate ($\mathbf{y}_{<t}$,$y_t$)'s score proportional to the probability of satisfying additional suffixes constraints $\mathbf{y}_{>t}$. In reality, we need only estimate two or three words in suffixes without estimating suffixes.

Intuitively, our lookahead heuristic for decoding brings two benefits. (1) The $y_t$ can be a token that

would satisfy a multi-token constraint or a phrase as the lookahead computes the average score ($y_t$,$\mathbf{y}_{>t}$). (2) When $y_t$ is one word in suffixes, the lookahead will help to decrease its score, thereby precluding it from being among the top $K$ tokens.

### 3.3 Substitute Ranking

After obtaining substitute candidates, existing LS methods (Zhou et al., 2019; Lacerra et al., 2021b; Seneviratne et al., 2022) obtain a contextualized representation of each substitute by replacing the target word with the substitute, and rank the substitutes by computing the cosine similarity of the target word vector with respect to that of each substitute. The similarity between the target word and the substitute does not provide sufficient information about whether the substitute will modify the sentence's meaning. After replacing the target word of the original sentence with the substitute to form the updated version, we attempt to evaluate the original sentence $\mathbf{x}$ and the updated sentence to rank the substitutes, as opposed to the target word and the substitute alone.

We formulate evaluating updated sentence as a text generation evaluation task. Assume that the updated sentence is denoted as $\mathbf{x}'$ after replacing the target word $x_t$ in $\mathbf{x}$ into one substitute. To accurately calculate a similarity score between $\mathbf{x}$ and $\mathbf{x}'$, we found BARTScore (Yuan et al., 2021) and BLEURT (Sellam et al., 2020) are specifically designed for text generation tasks, which aligns with the goal of lexical substitution. Therefore, they could be used to measure the quality of the substitutes.

BARTScore is a neural network-based evaluation metric that compares the likelihood of the original sentence and the updated sentence. It can assign higher scores to the sentences that are more likely to be original sentences. BLEURT is also a neural network-based evaluation metric, which is trained to predict how human-like a text is by comparing it with a large dataset of human-written texts. Those two metrics could assign a similarity or dissimilarity score, which allow the ranking of the substitutes based on how much similar to the original sentence they are, which might be better to rank the substitutes rather than other ranking methods (Zhou et al., 2019; Lacerra et al., 2021b; Seneviratne et al., 2022).

We have also incorporated the prediction scores of the substitute candidates generated by the paraphraser. Ultimately, our method employs a linear combination of the aforementioned three features (Paraphraser, BARTScore, BLEURT) to compute the final score for each substitute candidate.

## 4 Experiments

### 4.1 Experiment Setup

**LS Benchmarks.** Two widely used datasets, LS07 (McCarthy and Navigli, 2007) and CoInCo (Kremer et al., 2014), are chosen for the evaluation of LS methods. We also adopt the latest LS benchmark, Stanford Word Substitution Benchmark (SwordS) (Lee et al., 2021), which extends and improves CoInCo via crowdsourcing annotators in Amazon Mechanical Turk. Each instance in LS dataset is composed of a target word, its context, and corresponding substitutes. LS07 consists of 300 development examples and 1710 test instances for 201 polysemous words. CoInCo consists of 15K target instances with a given 35% development and 65% test. SwordS contains 762 test instances.

**Metrics.** For evaluating LS07 and CoInCo, we use the official metrics "best", "best-m", "oot", "oot-m" in SemEval 2007 task as well as Precision@1 (P@1) as our evaluation metrics, following the previous LS methods (Zhou et al., 2019; Michalopoulos et al., 2022). Among them, "best", "best-m" and "P@1" evaluate the quality of the best predictions, while both "oot" (out-of-ten) and "oot-m" evaluate the coverage of the gold substitute candidate list by the top 10 predictions.

In SwordS, a word is regarded as $acceptable$ if it is judged to be good by more than five out of ten annotators, and $conceivable$ if selected by at least one annotator. For the evaluation metrics, the authors (Lee et al., 2021) use the harmonic mean of the precision and recall given the gold and top-10 system-generated substitutes. As gold substitutes, they use either the acceptable or conceivable words, and calculate the corresponding scores $F_a$ and $F_c$, respectively.

**Baselines.** We compare our methods ParaLS and ParaLS$^\star$ with the following baselines, Word2Vec (Melamud et al., 2015b), BERT (Zhou et al., 2019), BERT+WordNet (Michalopoulos et al., 2022), GRoBERT (Lin et al., 2022), and XLNet+Word2Vec (Arefyev et al., 2020; Seneviratne et al., 2022). Arefyev et al. (Arefyev et al., 2020) linearly combine the prediction of pretrained language models XLNet and Word2Vec. Afterward, Seneviratne et

| Data set | Method | best | best-m | oot | oot-m | P@1 |
|---|---|---|---|---|---|---|
| LS07 | Word2Vec | 12.7 | 21.7 | 36.4 | 52.0 | - |
| | BERT | 20.3 | 34.2 | 55.4 | 68.4 | 51.1 |
| | BERT+WordNet | 21.1(16.3) | 35.5(27.6) | 51.3(45.6) | 68.6(62.4) | 51.7 (40.8) |
| | GR-RoBERT | 23.1(19.4) | 39.7(33.2) | 57.6(52.8) | 76.3 (71.5) | 55.0(47.4) |
| | XLNet+Word2Vec | 23.3(21.3) | 40.9(37.8) | 56.3(55.04) | 74.8(73.9) | 55.9 (50.5) |
| | ParaLS (ours) | 23.5(20.0) | 41.5(34.4) | 59.0(52.4) | 77.9(68.9) | 56.9(48.4) |
| | ParaLS⋆ (ours) | **24.0(22.3)** | **42.2(39.0)** | **60.5(57.3)** | **79.3(76.1)** | **58.8(54.3)** |
| CoInCo | Word2Vec | 8.1 | 17.4 | 26.7 | 46.2 | - |
| | BERT | 14.5 | 33.9 | 45.9 | 69.9 | 56.3 |
| | BERT+WordNet | 14.0 (11.3) | 29.7(23.8) | 38.0(33.6) | 59.2(54.4) | 50.5(41.3) |
| | GR-RoBERT | 15.2(13.1) | 34.4(28.8) | 45.3(40.9) | 71.3(66.6) | 55.9(48.8) |
| | XLNet+Word2Vec | 16.4(15.1) | 35.8(33.0) | 46.9(45.1) | 73.0(71.9) | 57.3(52.6) |
| | ParaLS(ours) | 18.1(13.8) | 40.1(29.5) | 50.7(41.7) | 78.1(65.6) | 62.4(50.0) |
| | ParaLS⋆(ours) | **18.5(16.8)** | **41.0(35.4)** | **52.1(48.3)** | **79.5(75.0)** | **64.1(57.8)** |

Table 2: Evaluation results of substitute generation and substitute ranking on LS07 and CoInCo datasets. The scores in parentheses are only calculated by the substitutes from the substitute generation step. The baselines are Word2Vec (Melamud et al., 2015a), BERT (Zhou et al., 2019), BERT+WordNet (Michalopoulos et al., 2022), GR-RoBERT (Lin et al., 2022), and XLNet+WordVec (Arefyev et al., 2020; Seneviratne et al., 2022). Best values are bolded.

al. (Seneviratne et al., 2022) adopt four features to rank the substitutes generated by XLNet and Word2Vec.

**Implementation Details.** To implement an English paraphraser, we fine-tune BART-base[2] model in fairseq. The initial learning rate is set to $lr = 3 \times 10^{-5}$ and dropout is set to 0.1. We adopt the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. We choose an English paraphrase dataset ParaBank2 (Hu et al., 2019) to train the paraphraser. In our experiments, we duplicate all the samples by exchanging source sentence and target sentence. We use the BLEURT large model[3] for the calculation of BLEURT score. BARTScore fine-tuned on ParaBank2 can be downloaded here[4]. We use the LS07 dev set for tuning the hyperparameters in our model. The weights for the prediction score (Paraphraser), BARTScore, and BLEURT for ParaLS and ParaLS⋆ are 0.02, 1, and 1, respectively. The number of outputted paraphrases $K$ is set to 50. The lookahead length of ParaLS⋆ is 2. Following the existing work (Zhou et al., 2019; Michalopoulos et al., 2022; Seneviratne et al., 2022), only the top 10 substitutes are used for evaluation.

### 4.2 Experimental Results

**Comparison of LS methods.** The results of our methods as well as the state-of-the-art methods on LS07 and CoInCo are displayed in Table 2. Typically, performance is evaluated by selecting the top substitutes after executing the substitute ranking step. To exclude the influence of substitute ranking, we also present the results without substitute ranking in parentheses.

As can be observed, our methods, ParaLS and ParaLS⋆, demonstrate superior performance over the latest LS methods (GR-RoBERT and XLNet+Word2Vec) across all metrics in the LS07 and CoInCo datasets. Notably, ParaLS⋆ without the step of substitute ranking outperforms all baselines, including the best baseline XLNet+Word2Vec, which utilizes four features for substitute ranking. ParaLS⋆ without substitute ranking significantly outperforms ParaLS without substitute ranking, which means that the decoding with lookahead heuristic in ParaLS⋆ is very useful.

| Method | $F_a$ | $F_c$ |
|---|---|---|
| GPT3 | 22.7 | 36.3 |
| WORDTUNE | 23.5 | 34.7 |
| BERT | 17.2 | 27.5 |
| XLNet+Word2Vec | 21.7(19.9) | 34.5(31.5) |
| ParaLS | 23.5 | 38.6 |
| ParaLS* | **24.9(22.8)** | **40.1(37.0)** |

Table 3: Results on SwordS dataset. The results of two commercial systems GPT3 (Brown et al., 2020) and WordTune (AI21, 2020) are from Lee et al. (Lee et al., 2021). For all metrics, the higher, the better.

[2]https://dl.fbaipublicfiles.com/fairseq/models/bart.base.tar.gz
[3]https://huggingface.co/Elron/bleurt-large-512
[4]https://github.com/neulab/BARTScore

| Method | LS07 | CoInCo |
|---|---|---|
| ParaLS⋆(Ours) | **65.2** | **60.0** |
| -w/o BARTScore | 63.6 | 59.1 |
| -w/o BLEURT | 64.1 | 58.9 |
| -w/o Paraphraser | 64.5 | 59.2 |
| o. Paraphraser | 61.9 | 57.5 |
| o. BARTScore | 62.8 | 57.4 |
| o. BLEURT | 59.5 | 55.3 |
| ParaLS(Ours) | 65.1 | 60.0 |
| XLNet+Word2Vec | 60.5 | 55.6 |
| BERT+WordNet | 60.6 | 58.0 |
| CILex3 | 57.8 | 53.6 |
| BERT | 58.6 | 55.2 |
| Word2Vec | 55.1 | 50.2 |

Table 4: Comparison of GAP scores (%) in the substitute ranking sub-task. The results from XLNet+Word2Vec (Arefyev et al., 2020), CILex3 (Seneviratne et al., 2022) are presented. "-w/o" indicates a ParaLS framework without the specific feature. "o." indicates that only one specific ranking feature is used.

The results on SwordS are presented in Table 3. Our method ParaLS and ParaLS⋆ achieve the highest $F_a$ score and $F_c$ score, largely outperforming the best baseline XLNet+Word2Vec as well as two commercial methods GPT-3 and WordTune. Unlike GPT-3, which is fine-tuned by a prompt-based learning framework from multiple samples of the development set in SwordS, ParaLS and ParaLS⋆ do not rely on any LS dataset.

In comparison to LS methods based on pre-trained language models, our methods possess the following three advantages:

(1) The paraphraser has been specifically trained to learn lexical variations. This could give it an advantage over pre-trained language models, which are generally trained on a wide range of tasks and may not be as focused on lexical substitution.

(2) The paraphraser is better at preserving the original meaning and context of the text, as it has been specifically designed to rewrite text while maintaining its meaning. This could be particularly important for lexical substitution tasks, as the goal is often to find substitutions that are semantically similar to the target word.

(3) The paraphraser can generate more diverse or varied substitutions. Pre-trained language models, on the other hand, are more general-purpose and may not be as adept at generating diverse substitutions.

**Comparison of Substitute Ranking.** We also evaluate our substitute ranking strategies on both the LS07 and CoInCo datasets. In this sub-task of LS task, assume that the substitute candidates are provided, each method aims to create the most appropriate ranking of the candidates. Following prior work (Zhou et al., 2019; Michalopoulos et al., 2022), we use GAP score [5] for evaluation in the subtask, which is a variant of MAP (Mean Average Precision). We also output the results of the proposed method ParaLS⋆ by removing one feature or two features.

The results are displayed in Table 4. XLNet+Word2Vec, BERT+WordNet, and CILEX3 utilize 2, 4, and 4 features respectively to rank the substitutes, which include Gloss-sentence similarity score, sentence similarity score, and WordNet similarity score, among others. Our results obtained solely by using the BARTScore or Paraphraser feature surpass those of the baselines, with BARTScore exhibiting particularly strong performance. BLEURT also demonstrates superior performance when compared to CILEX3 and BERT. These results confirm that text generation evaluation metrics (BARTScore or BLEURT) are better suited for substitute ranking than prior methods. The performance of ParaLS⋆ when one feature is removed demonstrates that all the features have a positive impact on the performance of ParaLS⋆.

The proposed strategy using BARTScore or BLEURT for ranking substitutes based on the change of the sentence's meaning after embedding them into the original sentences is likely effective because it directly addresses the primary goal of lexical substitution, which is to preserve the meaning of the original sentence while replacing a word. By using text generation evaluation metrics such as BARTScore and BLEURT to compute the relationship between the original and updated sentences, the method can quantify the extent to which the meaning of the original sentence has been preserved by each substitute.

**Ablation Study.** To further evaluate the impact of each ranking feature on the performance of our method, we conducted an ablation study on ParaLS⋆. The results are presented in Table 5. Both BARTScore and BLEURT are observed to be beneficial in enhancing the performance of ParaLS⋆. The ablation study, by isolating and testing the performance of individual features, illustrates that the Paraphraser feature alone achieves the best perfor-

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

| | best | b.m | oot | o.m | P@1 |
|---|---|---|---|---|---|
| ParaLS⋆ | **18.5** | **41.0** | **52.1** | **79.5** | **64.1** |
| -w/o Pa. | 17.8 | 39.7 | 51.4 | 79.0 | 61.2 |
| -w/o BA. | 17.83 | 39.0 | 51.3 | 78.1 | 62.1 |
| -w/o BL. | 17.4 | 38.2 | 50.2 | 77.8 | 60.6 |
| o. Pa. | 16.4 | 35.6 | 48.3 | 75.1 | 57.9 |
| o. BA. | 15.7 | 34.4 | 48.2 | 75.6 | 55.3 |
| o. BL. | 14.9 | 31.3 | 48.2 | 74.3 | 53.4 |

Table 5: Ablation study of ranking features for ParaLS⋆ on CoInco dataset. "-w/o" indicates ParaLS⋆ without the specific feature. "o." indicates that only one specific ranking feature is used. "Pa.", "BA.", "BL.", "b.m" and "o.m" are denoted as "Paraphrase", "BARTScore", "BLEURT", "Best.m" and "oot.m", respectively.

mance, thereby highlighting the effectiveness of our decoding with lookahead heuristics in generating high-quality substitutes.

**Case Study.** To quantitatively evaluate the effectiveness of the substitutes generated by LS methods, we present five instances of CoInCo for analysis. Table 6 displays the top five generated substitutes. Upon examination, we find that many suitable substitutes, marked in blue, are not present in the Labels. As the labels are human-annotated, it is not possible to provide all suitable substitutes for each target word. We posit that the actual performance of ParaLS and ParaLS⋆ is superior to the results computed by the metrics.

Furthermore, we see that our methods generate more high-quality substitutes than the baselines. Even when the methods generate unsuitable substitutes, the changes to the semantic information of the sentence are minimal. In the future, our methods could be utilized to enhance the coverage of substitutes in existing LS datasets.

## 5 Conclusions

We introduce two novel paraphraser-based LS methods named ParaLS and ParaLS⋆, which generate substitute candidates by considering the context and preserving the sentence's meaning. Specifically, we design two decoding strategies that center on lexical variations of the target word during decoding and propose a substitute candidate ranking strategy by utilizing the newest text generation evaluation metrics. Experimental results show that ParaLS and ParaLS⋆ significantly outperform the state-of-the-art LS methods. In the future, we will apply the methods to different languages, and verify our method on many downstream tasks to investigate

| | |
|---|---|
| **Inst1** | inauguration of **free** zone in · · · |
| Labels | open,unrestricted,unlimited, · · · |
| BERT | safe,open,public,reserve,reserved |
| XLNet | complimentar,open,exclusive, new,digital |
| ParaLS | open,liberty,fair,unrestricted, liberated |
| ParaLS⋆ | open,liberty,autonomous, independent,unrestricted |
| **Inst2** | i just hope they **keep** me here |
| Labels | retain,stash,leave,hold,guard,· · · |
| BERT | have,want,get,bring,take |
| XLNet | maintain,stay,hold,stick,have |
| ParaLS | hold,leave,stay,lock,have |
| ParaLS⋆ | hold,leave,have,stay,put |
| **Inst3** | · · · pulled out a secret **code** for · · · |
| Labels | encryption,signal,password, · · · |
| BERT | combination,key,sequence, message,number |
| XLNet | password,message,key, address,number |
| ParaLS | password,cipher,encryption, message,protocol |
| ParaLS⋆ | password,cipher,encryption, protocol,message |
| **Inst4** | · · · drop an **atomic** bomb · · · |
| Labels | nucleus,molecule,ion |
| BERT | bomb,element,atmosphere, nucleus,uranium |
| XLNet | earth,world,universe,planet,sun |
| ParaLS | nuclear,electron,nucleus, particle,bomb |
| ParaLS⋆ | nucleus,nuclear,bomb, electron,electrons |
| **Inst. 5** | i do it as **somebody** who who has a conscience · · · |
| Labels | someone,one,person,anyone,· · · |
| BERT | someone,anybody,person, anyone,somewhere |
| XLNet | someone,person,persons, somewhere,one |
| ParaLS | someone,one,anyone,person, anybody |
| ParaLS⋆ | someone,one,anyone,person, anybody |

Table 6: The top five substitutes of five instances in CoInCo by LS methods. The target word is bolded, the substitutes in labels are marked in red, and the suitable substitutes not in labels are marked in blue.

further the method's general availability.

## Limitations

Our method depends on a large-scale paraphrasing corpus. We only test our method on the English LS task. Excluding English, other languages have large-scale paraphrasing datasets available, e.g., French, German, Chinese, Spanish, etc. Our method can be easily extended to these languages. But, for some languages that cannot obtain enough paraphrasing datasets, our proposed method cannot be used. Another limitation is that our method may struggle to generate substitutions for rare or unusual words and phrases, as they may not have encountered sufficient examples of these words in the training paraphrase data.

## Ethics Statement

One potential ethical consideration related to a LS method based on a paraphraser is the potential for biased or unfair language generation. If the training data used to develop the paraphraser is biased in some way (e.g., it disproportionately represents certain groups of people or uses certain words and phrases in a biased manner), this could lead to biased substitutions being generated by the model. It is important to ensure that the training data used to develop the model is diverse and free of bias in order to minimize the potential for unfair or biased language generation.

Another ethical consideration is the potential for the LS method to be used for malicious purposes, such as creating fake or misleading content. It is important to consider the potential consequences of the LS method's outputs and to put safeguards in place to prevent the LS method from being used for nefarious purposes.

## Acknowledgement

This research is partially supported by the National Natural Science Foundation of China under grants 62076217 and 61906060, and the Blue Project of Yangzhou University.

## References

AI21. 2020. Wordtune (accessed 2020 oct 30). *https://www.wordtune.com/*.

Nikolay Arefyev, Boris Sheludko, Alexander Podolskiy, and Alexander Panchenko. 2020. A comparative study of lexical substitution approaches based on neural language models. *arXiv preprint arXiv:2006.00031*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Kyunghyun Cho. 2016. Noisy parallel approximate decoding for conditional recurrent language model. *arXiv preprint arXiv:1605.03835*.

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

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *NAACL-HLT*, pages 758–764.

Peter E Hart, Nils J Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.

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.

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.

J. Edward Hu, Abhinav Singh, Nils Holzenberger, Matt Post, and Benjamin Van Durme. 2019. Large-scale, diverse, paraphrastic bitexts via sampling and clustering. In *CoNLL*, pages 44–54, Hong Kong, China. Association for Computational Linguistics.

Sora Kadotani, Tomoyuki Kajiwara, Yuki Arase, and Makoto Onizuka. 2021. Edit distance based curriculum learning for paraphrase generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 229–234, Online. Association for Computational Linguistics.

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.

Reno Kriz, Eleni Miltsakaki, Marianna Apidianaki, and Chris Callisonburch. 2018. Simplification using paraphrases and context-based lexical substitution. In *NAACL*, pages 207–217.

Ilia Kulikov, Alexander H Miller, Kyunghyun Cho, and Jason Weston. 2019. Contextualized perturbation for textual adversarial attack. In *In Proceedings of the 12th International Conference on Natural Language Generation*, pages 76–87.

Caterina Lacerra, Tommaso Pasini, Rocco Tripodi, and Roberto Navigli. 2021a. Alasca: an automated approach for large-scale lexical substitution. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 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.

Mina Lee, Chris Donahue, Robin Jia, Alexander Iyabor, and Percy Liang. 2021. Swords: A benchmark for lexical substitution with improved data coverage and quality. In *NAACL*, pages 4362–4379, Online. Association for Computational Linguistics.

Yu Lin, Zhecheng An, Peihao Wu, and Zejun Ma. 2022. Improving contextual representation with gloss regularized pre-training. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 907–920, Seattle, United States. Association for Computational Linguistics.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States. Association for Computational Linguistics.

Diana McCarthy. 2002. Lexical substitution as a task for wsd evaluation. In *Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions*, pages 89–115.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *In Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 48—-53.

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

Oren Melamud, 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.

Yuxian Meng, Xiang Ao, Qing He, Xiaofei Sun, Qinghong Han, Fei Wu, Chun Fan, and Jiwei Li. 2021. ConRPG: Paraphrase generation using contexts as regularizer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2551–2562, Online and Punta Cana, Dominican Republic. 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.

Gustavo H Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In *AAAI*, pages 3761–3767.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple ppdb: A paraphrase database for simplification. In *ACL*, pages 143–148.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *ACL*, pages 425–430.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, Yang Shi, and Xindong Wu. 2021a. Lsbert: Lexical simplification based on bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3064–3076.

Jipeng Qiang, Xinyu Lv, Yun Li, Yunhao Yuan, and Xindong Wu. 2021b. Chinese lexical simplification. *IEEE Transactions on Audio, Speech and Language Processing.*, 29:1819–1828.

Jipeng Qiang and Xindong Wu. 2021. Unsupervised statistical text simplification. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1802–1806.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

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.

Yik-Cheung Tam. 2020. Cluster-based beam search for pointer-generator chatbot grounded by knowledge. *Computer Speech & Language*, 64:101094.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *AAAI*.

John Wieting and Kevin Gimpel. 2017. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732*.

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 *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Deniz Yuret. 2007. Ku: Word sense disambiguation by substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 207–214.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. Bert-based lexical substitution. In *ACL*, pages 3368–3373.

# A Appendix A (More Experiments for Ablation Study)

**1. The baselines.** We compare our methods ParaLS and ParaLS⋆ with the following baselines.

(1) Word2Vec: The words that have the highest similarities are selected as substitute candidates from the word embedding modeling whose vectors are closer in terms of cosine similarity with the target word (Melamud et al., 2015b).

(2) BERT: BERT proposed by (Zhou et al., 2019) applies dropout to the embedding of the target word for partially obscuring the target word.

(3) BERT+WordNet: Michalopulos et al. (Michalopoulos et al., 2022) integrated the knowledge from WordNet into the embedding of BERT.

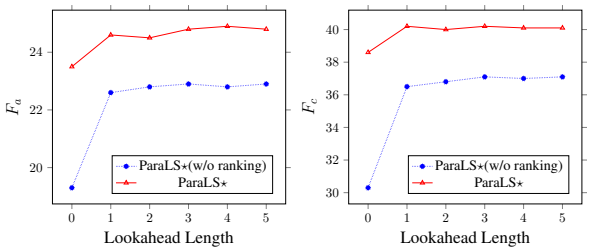
(4) GR-RoBERT: Lin et al. (Lin et al., 2022) proposed an auxiliary gloss regularizer module to BERT pre-training, to enhance word semantic similarity.

(5) XLNet+Word2Vec(Arefyev et al., 2020; Seneviratne et al., 2022): (Arefyev et al., 2020) linearly combines the prediction of pretrained language models XLNet and Word2Vec. Afterward, Seneviratne et al. (Seneviratne et al., 2022) adopt four features to rank the substitutes generated by XLNet and Word2Vec.

**2. Influence of different ranking features.** In the paper, we give the results of CoInCo datasets. Here, we give the results of LS07 datasets. The results are shown in Table 7. The conclusions are consistent with the conclusions of CoInCo.

| | best | b.m | oot | o.m | P@1 |
|---|---|---|---|---|---|
| ParaLS⋆ | **24.0** | **42.0** | **60.5** | **79.3** | **58.8** |
| -w/o Pa. | 22.2 | 38.9 | 58.5 | 76.8 | 54.4 |
| -w/o BA. | 23.6 | 40.9 | 59.6 | 78.0 | 57.3 |
| -w/o BL. | 23.7 | 41.4 | 59.1 | 78.5 | 58.0 |
| o. Pa. | 22.3 | 39.0 | 57.3 | 76.1 | 54.3 |
| o. BA. | 20.2 | 35.0 | 55.8 | 75.3 | 50.5 |
| o. BL. | 18.6 | 30.0 | 54.9 | 70.7 | 46.7 |

Table 7: Ablation study of ranking features for ParaLS⋆ on LS07 dataset. "-w/o" indicates ParaLS⋆ without the specific feature. "o." indicates that only one specific ranking feature is used.




(a) (b)

Figure 2: Effect of varying lookahead length for ParaLS⋆. (a) the results using metric $F_a$, and (b) the results using metric $F_c$.

**3. Influence of lookahead length.** ParaLS⋆ has a parameter of lookahead length. In this experiment, we will analyze the influence of lookahead length on the performance of ParaLS⋆. We vary the length of the lookahead from 0 to 5. When lookahead length equals 0, ParaLS⋆ is transformed into ParaLS.

The results are displayed in Figure 2. We see that the performance ParaLS⋆ is robust when varying lookahead length.

**4. The running time of LS method.** We give the average running time of one instance in Table 8. We run 100 instances in CoInCo dataset, and compute the average time of one instance.

We see that ParaLS only need 1.05 second for one instance, close to BERT(Zhou et al., 2019). XLNet+Word2Vec (Seneviratne et al., 2022) is the slowest LS method.

**5. Influence of different paraphrasers.** We do these experiments to verify the influence of different paraphrasers on the performance of ParaLS. In our paper, we adopt pretrained modeling BART to fine-tune an English paraphraser. Here, we train a Transformer model in FairSeq with a 6-layer encoder and decoder, 512-dimensional embeddings,

| Method | Runtime(s) |
|---|---|
| BERT | 1.00 |
| XLNet+Word2Vec | 3.56 |
| ParaLS | 1.05 |
| ParaLS* w/o ranking | 1.6 |
| ParaLS* | 1.96 |

Table 8: The average running time of one instance.

8 encoder-decoder attention heads, and 0.1 dropout. The initial learning rate is set to $lr = 3 \times 10^{-4}$. We adopt the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.

The results are shown in Table 9. We see that the performance of our proposed ParaLS and ParaLS$\star$ is not significantly affected by the specific paraphrase model that is used.

## B Appendix B (Case Study)

Here, we give the generated top 10 substitutes of 10 instances in CoInCo to analyze the generated substitutes by our method (ParaLS and ParaLS$\star$) and the baselines (BERT (Zhou et al., 2019) and XLNet (Seneviratne et al., 2022).

| Dataset | Method | best | best-m | oot | oot-m | P@1 |
|---|---|---|---|---|---|---|
| LS07 | XLNet+Word2Vec | 23.3 | 40.9 | 56.3 | 74.8 | 55.9 |
| | ParaLS (Transformer) | **24.1** | **42.4** | 58.2 | 76.5 | 58.3 |
| | ParaLS⋆(Transformer) | **24.1** | **42.2** | 59.4 | 77.4 | 58.6 |
| | ParaLS(BART) | 23.5 | 41.5 | 59.0 | 77.9 | 56.9 |
| | ParaLS⋆(BART) | 24.0 | 42.2 | **60.5** | **79.3** | **58.8** |
| CoInCo | XLNet+Word2Vec | 16.4 | 35.8 | 46.9 | 73.0 | 57.3 |
| | ParaLS(Transformer) | 18.1 | 40.0 | 49.2 | 75.4 | 62.6 |
| | ParaLS⋆(Transformer) | 18.2 | 40.4 | 50.3 | 76.7 | 63.3 |
| | ParaLS(BART) | 18.1 | 40.1 | 50.7 | 78.1 | 62.4 |
| | ParaLS⋆(BART) | **18.46** | **40.96** | **52.14** | **79.48** | **64.11** |

Table 9: Results of ParaLS and ParaLS⋆ using two different paraphrasers on LS07 and CoInCo datasets. For comparison, we also show the results of the best baseline "XLNet+Word2Vec" (Seneviratne et al., 2022).

| | |
|---|---|
| **Inst. 1** | Chron editors note : each week , the chronicle offers readers a look at the more unusual fruits, **vegetables** and herbs of each season and how to use them . |
| Labels | veggies;produce;vegetable specimen;plant;herbage; |
| BERT | foods;spices;grains;crops;berries;grasses;beans;plants;shrubs;ingredients |
| XLNet | herbs;crops;flowers;onions;foods;plants;grains;seeds;fruits;potatoes |
| ParaLS | greens;plants;foodstuffs;crops;veggies;veg;seeds;varieties;vines;cereals |
| ParaLS⋆ | greens;plants;crops;veggies;varieties;foodstuffs;seeds;veg;produce;vines |
| **Inst. 2** | If they continued to **resist** , he pulled out a secret code for their bosses . |
| Labels | refuse;rebel;thwart the matter;stonewall;refrain;oppose;object;disbelieve;defy;decline; counteract;be uncooperative;abstain; |
| BERT | refuse;struggle;protest;rebel;submit;comply;obey;reject;flee;escape |
| XLNet | refuse;protest;persist;comply;react;hesitate;evade;obey;respond;submit |
| ParaLS | refuse;oppose;protest;struggle;fight;evade;rebel;object;deny;revolt |
| ParaLS⋆ | refuse;oppose;fight;struggle;protest;evade;rebel;object;deny;defy |
| **Inst. 3** | He 's a right handed bat , which complements Palmeiro off the **bench** . |
| Labels | wood;wait area;stand;seat;reserve;replacement;relief;pine;dugout;box;bleacher;backup; auxiliary |
| BERT | field;pitch;team;plate;bat;opener;start;rest;ball;squad |
| XLNet | lineup;mound;team;roster;plate;field;diamond;box;spot;bullpen |
| ParaLS | stand;field;court;bleachers;dugout;plate;pitch;mound;ground;line |
| ParaLS⋆ | field;stand;court;pitch;deck;box;dugout;plate;mound;bleachers |
| **Inst. 4** | Grande dame of cooking still going strong at 90 : Julia Child **celebrates** in san francisco |
| Labels | rejoice;party;enjoy;dance |
| BERT | celebrations;celebration;sings;remembers;promotes;holidays;performs;starts;wins;promotions |
| XLNet | celebration;celebrations;holidays;festivities;holiday;feast;birthday;shows;parade;festival |
| ParaLS | commemorates;is;presents;gala;festivities;commemorate;anniversary;party;birthday;glorifies |
| ParaLS⋆ | commemorates;rejoice;feast;rejoices;feasts;cheers;festivities;dances;revels;presents |
| **Inst. 5** | Responsible seafood **sales** are the catch of the day |
| Labels | purchase;transaction;vending;purchasing;deal;buying;barter |
| BERT | purchases;selling;markets;prices;vendors;trading;buyers;stores;products;donations |
| XLNet | purchases;selling;sellers;markets;marketing;shipments;prices;buyers;businesses;retailers |
| ParaLS | purchases;sells;selling;exports;products;sold;prices;markets;sell;deals |
| ParaLS⋆ | purchases;sells;deliveries;exports;transactions;products;selling;supplies;markets;deals |

Table 10: The top 10 substitutes of five instances in CoInCo using LS methods. The target word is bolded, the substitutes in labels are marked in red, and the suitable substitutes not in labels are marked in blue. Here, the baselines are BERT (Zhou et al., 2019) and XLNet (Seneviratne et al., 2022). "XLNet+Word2Vec" is abbreviated as XLNet.

| | |
|---|---|
| **Inst. 6** | Sony corp. completed its tender offer for Columbia pictures entertainment inc., with Columbia shareholders **tendering** 99.3% of all common shares outstanding by the Tuesday deadline. |
| Labels | pay;offer;issue;give;get;earn;deal |
| BERT | bid;auction;submit;deposit;present;surrender;broker;ballot;forward;dispatch |
| XLNet | offering;bidding;submitting;selling;taking;placing;securing;buying;providing;accepting |
| ParaLS | offering;submitting;bidding;accepting;bid;requesting;receiving;giving;providing;buying |
| ParaLS⋆ | offering;bidding;submitting;bid;accepting;giving;proposing;providing;requesting;holding |
| **Inst. 7** | " We 've discontinued **selling** swordfish , chilean seabass , orange roughy and marlin , " |
| Labels | offering;vend;serve;peddling;distributing |
| BERT | marketing;offering;buying;producing;sales;retail;sale;trading;shipping;export |
| XLNet | buying;sales;marketing;sale;offering;purchasing;trading;promoting;shipping;supplying |
| ParaLS | retailing;marketing;trading;distributing;offering;sellin;sales;trafficking;sale;serving |
| ParaLS⋆ | marketing;trading;retailing;distributing;buying;offering;supplying;delivering;peddling; merchanting |
| **Inst. 8** | The federal complaint offers many **details** of the alleged conspiracy , including excerpts from a transcript of the Italian wiretaps . |
| Labels | specific;point;fact;tidbit;snippet;item;issue;facet;count;account |
| BERT | outlines;information;descriptions;specifications;highlights;documents;features;stories;facts; terms |
| XLNet | descriptions;information;outlines;elements;aspects;facts;highlights;accounts;components; features |
| ParaLS | particulars;aspects;specifics;facts;information;evidence;elements;indications;clarifications; facets |
| ParaLS⋆ | particulars;aspects;specifics;descriptions;facts;elements;evidence;indications;information; facets |
| **Inst. 9** | The **new** factory , which will begin normal production early next year , will employ about 1,000 people . |
| Labels | late;most recent;recent;projected;pristine;future;fresh;expect;come;added |
| BERT | rebuilt;expanded;upcoming;planned;expanding;proposed;combined;large;larger;second |
| XLNet | future;modern;latest;proposed;first;planned;expanded;current;original;main features |
| ParaLS | fresh;young;fellow;rookie;incoming;recent;next;emerging;younger;own |
| ParaLS⋆ | next;fresh;latest;future;emerging;recent;novel;innovative;production;construction |
| **Inst. 10** | Electronic theft by foreign and **industrial** spies and disgruntled employees is costing U. S. companies billions and eroding their international competitive advantage . |
| Labels | business;trade;mechanized;manufacturing;industrialized;economic |
| BERT | industry;manufacturing;corporate;commercial;technical;multinational;technological;factory; internal;chemical |
| XLNet | industry;corporate;domestic;commercial;internal;institutional;international;regional;national independent; |
| ParaLS | commercial;manufacturing;corporate;factory;business;economic;sectoral;professional; technological;international |
| ParaLS⋆ | manufacturing;commercial;business;corporate;factory;professional;economic;sectoral; technological;international |

Table 11: The top 10 generated substitutes of five instances in CoInCo using LS methods. The target word is bolded, the substitutes in labels are marked in red, and the suitable substitutes not in labels are marked in blue. Here, the baselines are BERT (Zhou et al., 2019) and XLNet (Seneviratne et al., 2022). "XLNet+Word2Vec" is abbreviated as XLNet.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*Limitations*

☑ A2. Did you discuss any potential risks of your work?
*Ethics Statement*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*Abstract and Introduction*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☒ Did you use or create scientific artifacts?

*Left blank.*

☐ B1. Did you cite the creators of artifacts you used?
*No response.*

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*No response.*

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*No response.*

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*No response.*

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*No response.*

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*No response.*

### C ☑ Did you run computational experiments?

*3*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*3*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*3*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*3*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*3*

**D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*No response.*

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*No response.*

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*No response.*

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*No response.*

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*No response.*