

MBZUAI

Digital.Commons@MBZUAI

---

Natural Language Processing Faculty  
Publications

Scholarly Works

---

9-17-2022

## Unsupervised Lexical Substitution with Decontextualised Embeddings

Takashi Wada

Timothy Baldwin

Yuji Matsumoto

Jey Han Lau

Follow this and additional works at: <https://dclibrary.mbzuai.ac.ae/nlpfp>



Part of the [Artificial Intelligence and Robotics Commons](#)

Preprint: arXiv

Archived with thanks to arXiv

Preprint License: CC by 4.0

Uploaded 12 October 2022

---

# Unsupervised Lexical Substitution with Decontextualised Embeddings

Takashi Wada<sup>1,2</sup> Timothy Baldwin<sup>1,3</sup> Yuji Matsumoto<sup>2</sup> Jey Han Lau<sup>1</sup>

<sup>1</sup> School of Computing and Information Systems, The University of Melbourne

<sup>2</sup> RIKEN Center for Advanced Intelligence Project (AIP)

<sup>3</sup> Department of Natural Language Processing, MBZUAI

twada@student.unimelb.edu.au, tb@ldwin.net

yuji.matsumoto@riken.jp, jeyhan.lau@gmail.com

## Abstract

We propose a new unsupervised method for lexical substitution using pre-trained language models. Compared to previous approaches that use the generative capability of language models to predict substitutes, our method retrieves substitutes based on the similarity of contextualised and *decontextualised* word embeddings, i.e. the average contextual representation of a word in multiple contexts. We conduct experiments in English and Italian, and show that our method substantially outperforms strong baselines and establishes a new state-of-the-art without any explicit supervision or fine-tuning. We further show that our method performs particularly well at predicting low-frequency substitutes, and also generates a diverse list of substitute candidates, reducing morphophonetic or morphosyntactic biases induced by article–noun agreement.<sup>1</sup>

## 1 Introduction

There has been growing interest in developing automatic writing support systems to assist humans to write documents. One relevant task to this research goal is lexical substitution, where given a target word and its surrounding context, a system suggests a list of word substitutions that can replace the target word without changing its core meaning. For instance, given the target word *great* and the context *He is a great artist*, the model might suggest alternative words such as *outstanding*, *terrific*, or *distinguished*. Writers can use such suggestions to improve the fluency of their writing, reduce lexical repetition, or search for better expressions that represent their ideas more creatively.

As with other NLP tasks, recent studies have shown that masked language models such as BERT (Devlin et al., 2019) perform very well on lexical substitution, even without any task-specific supervision. A common approach is to employ language

models as generative models and predict substitutes based on their generative capability. However, this approach has some limitations. First, it is extremely difficult for language models to predict rare words — especially those that are segmented into multiple subword tokens — since the models inevitably assign them very low probabilities. Second, word prediction is highly affected by morphosyntactic constraints from the surrounding context, which overshadows the (arguably more important) question of semantic fit. For instance, if the target word is *increase* in the context ... *with an increase in ...*, language models tend to suggest words that also start with a vowel sound due to the presence of *an* before the target word, missing other possible substitutes such as *hike* or *boost*. In fact, this problem is even more pronounced in languages where words have grammatical gender (e.g. Italian nouns) or a high degree of inflection (e.g. Japanese verbs).

In this paper, we propose a new approach that explicitly deals with these limitations. Instead of generating words based on language model prediction, we propose to find synonymous words based on the similarity of contextualised and *decontextualised* word embeddings, where the latter refers to the “average” contextual representation of a word in multiple contexts. Experiments on English and Italian lexical substitution show that our fully unsupervised method outperforms previous models by a large margin. Furthermore, we show that our model performs particularly well at predicting low-frequency words, and also generates more diverse substitutes with less morphophonetic or morphosyntactic bias, e.g. as a result of article–noun agreement in English and Italian.

## 2 Method

### 2.1 Our Approach

Given a sentence that contains a target word  $x$  and its surrounding context  $c$ , we first feed the sentence

<sup>1</sup>Code is available at: [https://github.com/twadada/lexsub\\_decontextualised](https://github.com/twadada/lexsub_decontextualised).

into a pre-trained transformer model (Vaswani et al., 2017) such as BERT and generate the contextualised representations of  $x$ :  $f^\ell(x, c) \in \mathbb{R}^d$ , where  $\ell (\leq L)$  denotes the layer of the model. We propose to predict substitutes of  $x$  by retrieving words that have similar representations to  $f^\ell(x, c)$ . To this end, we calculate  $S(y|x, c)$ : the score of  $y$  being a substitute of  $x$  in context  $c$ , as follows:

$$\begin{aligned} S(y|x, c) &= \cos(f(y), f(x, c)), & (1) \\ f(x, c) &= \sum_{\ell \in Z} f^\ell(x, c), \\ f(y) &= \frac{1}{N} \sum_i \sum_{\ell \in Z} f^\ell(y, c'_i), \end{aligned}$$

where  $f(y) \in \mathbb{R}^d$  denotes the *decontextualised* word embedding of  $y$ ;  $Z$  is a set of selected layers; and  $\cos(a, b)$  denotes the cosine similarity between  $a$  and  $b$ . To obtain  $f(y)$ , we randomly sample  $N$  sentences ( $c'_1, c'_2, \dots, c'_N$ ) that contain  $y$  from a monolingual corpus, and take the average of the contextualised representations of  $y$  given  $c'_i$ :  $f^\ell(y, c'_i)$ . We pre-compute  $f(y)$  for each word  $y$  in our pre-defined vocabulary  $\tilde{V}$ , which consists of *lexical items* (i.e. no subwords) and contains different words from the pretrained model’s original vocabulary  $V$ . If  $y$  is segmented into multiple subwords (using the pretrained model’s tokeniser), we average its subword representations — this way we can include low-frequency words in  $\tilde{V}$  and generate diverse substitutes. We obtain  $f(x, c)$  and  $f(y)$  by summing representations across different layers  $\ell \in Z$  to capture various lexical information.<sup>2</sup>

## 2.2 Multi-Sense Embeddings

Representing  $f(y)$  in Eqn. (1) as a simple average of the contextualised representations of  $y$  is clearly limited when  $y$  has multiple meanings, since the representations will likely vary depending on its usage. For instance, Wiedemann et al. (2019) show that BERT representations of polysemous words such as *bank* create distinguishable clusters based on their usages. To address this issue, we first group the  $N$  sentences into  $K$  clusters based on the usages of  $y$ , and for each cluster  $k$ , we obtain the decontextualised embedding  $f^k(y)$  by averaging the contextualised representations, i.e.,

$$f^k(y) = \frac{1}{|C^k|} \sum_{c' \in C^k} \sum_{\ell \in Z} f^\ell(y, c'),$$

<sup>2</sup>We also tried taking the weighted sum of the different-layer embeddings, but we did not see noticeable improvement.

where  $C^k$  denotes the set of the sentences that belong to the cluster  $k$ . To obtain clusters, we apply  $K$ -means (Lloyd, 1982; Arthur and Vassilvitskii, 2007) to the L2-normalised representations of  $y$  in  $N$  sentences.<sup>3</sup> We expect that if  $y$  has multiple senses,  $f^k(y)$  will to some degree capture the different meanings.<sup>4</sup> This methodology has been shown to be effective by Chronis and Erk (2020) on context-independent word similarity tasks. With  $f^k(y)$ , we can refine the similarity score  $S(y|x, c)$  in Eqn. (1) as follows:

$$S(y|x, c) = \max_k \cos(f^k(y), f(x, c)).$$

In this way, we can compare  $x$  with  $y$  based on the sense that is most relevant to  $x$ . Furthermore, we capture *global similarity* between  $x$  and  $y$  as:

$$S(y|x, c) = \max_k \lambda \cos(f^k(y), f(x, c)) + (1 - \lambda) \cos(f^k(y), f^{j_c}(x)), \quad (2)$$

$$j_c = \operatorname{argmax}_j \cos(f^j(x), f(x, c)), \quad (3)$$

where the second term in Eqn. (2) corresponds to the global similarity, which compares  $x$  and  $y$  outside of context  $c$ .<sup>5</sup> However, it still considers  $c$  in Eqn. (3) to retrieve the cluster that best represents the meaning of  $x$  given  $c$ .

While Eqn. (2) generally generates high-quality substitutes, we found that it sometimes retrieves words that share the same root word as  $x$  and yet do not make good substitutes (e.g. *pay* and *payer*). This is mainly due to the fact that the vocabulary  $\tilde{V}$  contains a large number of derivationally-related words, some of which are out-of-vocabulary (OOV) in the original vocabulary  $V$  (e.g. *pay* ##*er*). To address this problem, we add a simple heuristic where  $y$  is discarded if the normalised edit distance<sup>6</sup> between  $x$  and  $y$  is less than a threshold (0.5 for our English and Italian experiments).<sup>7</sup>

## 2.3 Reranking

In Eqn. (2), the context  $c$  affects the representation of  $x$  but not  $y$ . Ideally, however, we want to con-

<sup>3</sup>We concatenate  $f^\ell(y, c)$  across multiple layers  $\ell \in Z$ .

<sup>4</sup>Note that the number of clusters  $K$  is fixed across all words, forcing the model to “split” and “lump” senses (Hanks, 2012) to varying degrees.

<sup>5</sup>To obtain  $f^j(x)$ , we compute the decontextualised embedding of  $x$  and apply  $K$ -means, as we do to compute  $f^k(y)$ . When  $x$  is not included in our pre-defined vocabulary  $\tilde{V}$ , we set  $\lambda$  to 1 and ignore the second term in Eqn. (2).

<sup>6</sup>The distance normalised by the maximum string length.

<sup>7</sup>We tuned this threshold based on English development data (i.e. the development split of SWORDS).

sider the context  $c$  on both sides to find the words that best fit the context. Therefore, we first generate top- $M$  candidates based on Eqn. (2), and rerank them using the following score:

$$S(y|x, c) = \frac{1}{|Z|} \sum_{\ell \in Z} \cos(f^\ell(y, c), f^\ell(x, c)), \quad (4)$$

where  $f^\ell(y, c)$  denotes the contextualised representation of  $y$  given  $c$ , which can be obtained by replacing  $x$  in  $c$  with  $y$  and feeding it into the model. In Eqn. (4), we calculate the similarity at each layer  $\ell \in Z$  and take the average, which yields small yet consistent improvements over averaging the embeddings first and then calculating the similarity.<sup>8</sup> We limit the use of this scoring method to the  $M$  candidates only, since it is computationally expensive to calculate  $f^\ell(y, c)$  for every single word  $y$  in  $\tilde{V}$ . Previously, a similar method was employed by Lee et al. (2021) but they used the last layer only (i.e.  $Z = \{L\}$ ). We show that using multiple layers substantially improves the performance. Following Lee et al. (2021), we set  $M$  to 50.

## 2.4 Comparison to Previous Approaches

Our approach contrasts with previous approaches (Zhou et al., 2019; Lee et al., 2021; Yang et al., 2022) that employ BERT as a generative model and predict lexical substitutes based on the generation probability  $P(y|x, c)$ :

$$P(y|x, c) = \frac{\exp(E_y f^{\hat{L}}(x, c) + b_y)}{\sum_{\hat{y} \in V} \exp(E_{\hat{y}} f^{\hat{L}}(x, c) + b_{\hat{y}})}, \quad (5)$$

where  $E_y \in \mathbb{R}^d$  denotes the output embedding of  $y$ , which is usually tied with the input word embedding;  $f^{\hat{L}}(x, c)$  is the representation at the very last layer of the model;<sup>9,10</sup> and  $b_y$  is a scalar bias. While this approach is straightforward and well motivated, its predictions are highly influenced by morphosyntax, as discussed in Section 1. Moreover, Eqn. (5) shows three additional limitations compared to our approach: (1) the prediction is conditioned on the last layer only, despite previous

<sup>8</sup>In Eqn. (1), we obtained similar results by averaging the embeddings or cosine similarities across layers.

<sup>9</sup>Note that this does not always correspond to the last layer of transformer:  $f^L(x, c)$ . E.g., BERT calculates  $f^{\hat{L}}(x, c)$  by applying a feed forward network and layer normalisation to  $f^L(x, c)$ , whereas for XLNET,  $f^{\hat{L}}(x, c) = f^L(x, c)$ .

<sup>10</sup>When  $x$  consists of multiple subwords, the representation of the first or longest token is usually used.

studies showing that different layers capture different information, with the last layer usually containing less semantic information than the lower or middle layers (Bommasani et al., 2020; Tenney et al., 2019); (2)  $y$  is represented by the single vector  $E_y$ , which may not work well when  $y$  has multiple meanings — we alleviate this by clustering the embeddings (Section 2.2); and (3) the model is not capable of generating OOV words, unless we force the model to decode multiple subwords, e.g. by using multiple mask tokens or duplicating  $x$ . Our approach, in contrast, can include rare words in the pre-defined vocabulary  $\tilde{V}$  and generate diverse substitutes (Section 4.4).

## 3 Experiments

### 3.1 Data and Evaluation

We conduct experiments in two evaluation settings: generation and ranking. In the generation setting, systems produce lexical substitutes given target words and sentences, while in the ranking setting, they are also given substitute candidates and rank them based on their appropriateness.

For the generation task, we base our experiments on SWORDS (Lee et al., 2021), the largest English lexical substitution dataset, which extends and improves CoInCo (Kremer et al., 2014) by introducing a new annotation scheme: in CoInCo, the annotators were asked to come up with substitutes by themselves, whereas in SWORDS, the annotators were given substitute candidates pre-retrieved from a thesaurus, and only had to make binary judgements (“good” or “bad”).<sup>11</sup> 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. In this way, SWORDS provides more comprehensive lists of substitutes, including many low-frequency words that are good substitutes and yet difficult for humans to suggest — these words are of particular interest to us. For the evaluation metrics, the authors use the harmonic mean of the precision and recall given the gold and top-10 system-generated substitutes.<sup>12</sup> As gold substitutes, they use either the acceptable or conceivable words, and calculate the corresponding scores  $F_a$  and  $F_c$ , respectively.

<sup>11</sup>The annotators were asked if they would consider using the substitute candidate to replace the target word as the author of the context.

<sup>12</sup>More precisely, their evaluation script lemmatises the top-50 substitutes first and then extracts the top-10 distinct words.

They also propose to measure those scores in both *strict* and *lenient* settings, which differ in that in the lenient setting, candidate words that are not scored under SWORDS are filtered out and discarded.

In the ranking task, we evaluate models on the traditional SemEval-2007 Task 10 (“SemEval-07”) data set (trial+test) (McCarthy and Navigli, 2007), as well as SWORDS. For the evaluation metric, we follow previous work in using Generalized Average Precision (GAP; Kishida (2005)):

$$GAP = \frac{\sum_{i=1}^N I(\alpha_i) p_i}{\sum_{i=1}^R I(\beta_i) \bar{\beta}_i}, \quad p_i = \frac{\sum_{k=1}^i \alpha_k}{i}, \quad (6)$$

where  $\alpha_i$  and  $\beta_i$  denote the gold weight of the  $i$ -th item in the predicted and gold ranked lists respectively, with  $N$  and  $R$  indicating their sizes;  $I(\alpha_i)$  is a binary function that returns 1 if  $\alpha_i > 0$ , and 0 otherwise; and  $\bar{\beta}_i$  is the average weight of the gold ranked list from the 1st to the  $i$ -th items. In our task, the weight corresponds to the aptness of the substitute, which is set to zero if it is not in the gold substitutes. Following previous work (Melamud et al., 2015; Arefyev et al., 2020), we ignore multiword expressions in SemEval-07.<sup>13</sup>

### 3.2 Models

As shown in Eqn. (2), our approach requires only the vector representations of words and hence is applicable to any text encoder model. Therefore, we test our method with various pre-trained models, including five masked language models: BERT (Devlin et al., 2019), mBERT, SpanBERT (Joshi et al., 2020), XLNet (Yang et al., 2019), and MPNet (Song et al., 2020); one encoder-decoder model: BART (Lewis et al., 2020); and two discriminative models: ELECTRA (Clark et al., 2020) and DeBERTa-V3 (He et al., 2021).<sup>14</sup> We also evaluate two sentence-embedding models: MPNet-based sentence transformer (Reimers and Gurevych, 2019) and SimCSE (Gao et al., 2021), both of which are fine-tuned on semantic downstream tasks such as MNLI and achieve good performance on sentence-level tasks. Finally, we also evaluate the encoder of the fine-tuned mBART on English-to-Many translation (Tang et al., 2021). Note that the discriminative models and embedding models (e.g. NMT-encoder, SimCSE) cannot generate words and hence are incompatible with the previous approach described in Section 2.4.

<sup>13</sup>We run the evaluation code at <https://github.com/orenmel/lexsub> with the *no-mwe* option.

<sup>14</sup>See Appendix A for the details of all models.

| Models                           | Lenient     |             | Strict      |             |
|----------------------------------|-------------|-------------|-------------|-------------|
|                                  | $F_a$       | $F_c$       | $F_a$       | $F_c$       |
| HUMANS                           | 48.8        | 77.9        | –           | –           |
| CoInCo                           | 34.1        | 63.6        | –           | –           |
| GPT-3                            | <b>34.6</b> | 49.0        | 22.7        | 36.3        |
| BERT-K <sup>15</sup>             | 32.4        | 55.4        | 19.2        | 30.4        |
| (w/o rerank) <sup>15</sup>       | 31.8        | 54.9        | 15.7        | 24.4        |
| BERT-M                           | 30.9        | 48.3        | 16.2        | 25.4        |
| (w/o rerank)                     | 30.9        | 48.1        | 10.7        | 16.5        |
| Zhou et al. (2019) <sup>15</sup> | 32.0        | 55.4        | 17.4        | 27.5        |
| Yang et al. (2022) <sup>16</sup> | 31.9        | 54.9        | 16.7        | 28.4        |
| OURS                             |             |             |             |             |
| BERT                             | 33.2        | 64.1        | 21.1        | 34.9        |
| (w/o rerank)                     | 33.0        | 63.8        | 20.7        | 34.4        |
| (w/o rerank, heuristic)          | 33.6        | 64.0        | 20.2        | 32.4        |
| mBERT                            | 27.0        | 52.7        | 12.4        | 22.6        |
| SpanBERT                         | 32.6        | 61.4        | 20.9        | 34.0        |
| MPNet                            | 33.8        | 63.8        | 22.0        | 34.1        |
| XLNet                            | 34.4        | 65.3        | 23.3        | 37.4        |
| ELECTRA                          | 33.5        | 64.2        | 23.2        | 36.7        |
| DeBERTa-V3                       | 33.6        | <b>65.8</b> | <b>24.5</b> | <b>39.9</b> |
| BART (Enc)                       | 33.6        | 62.8        | 21.9        | 34.8        |
| BART (Dec)                       | 33.5        | 60.5        | 21.4        | 34.0        |
| BART (Enc-Dec)                   | 33.7        | 64.9        | 23.5        | 37.2        |
| SBERT (MPNet)                    | <b>34.6</b> | 64.0        | 21.8        | 33.5        |
| SimCSE (BERT)                    | 33.4        | 64.3        | 21.6        | 35.7        |
| NMT (mBART)                      | 28.7        | 55.6        | 13.4        | 22.2        |
| OURS (Rank Candidates)           |             |             |             |             |
| XLNet                            | 35.2        | 72.9        | –           | –           |
| BART                             | 34.8        | 72.4        | –           | –           |
| DeBERTa-V3                       | 35.1        | 72.2        | –           | –           |

Table 1: The results for the generation task in the lenient and strict settings. The best scores are boldfaced.

We use the same vocabulary  $\tilde{V}$  for all models, which consists of the 30,000 most common words<sup>17</sup> in the OSCAR corpus (Ortiz Suárez et al., 2020). We set the number of sentences we sample from OSCAR to calculate the decontextualised embeddings, i.e.  $N$ , to 300; the clustering size  $K$  to 4; and  $\lambda$  in Eqn. (2) to 0.7. For the set of transformer layers,  $Z$ , we employ all layers except for the first and

<sup>15</sup>These results differ slightly from the reported scores in Lee et al. (2021), due to a bug in their code.

<sup>16</sup>Updated from the original scores by the authors after they fixed some critical issues in their evaluation setup.

<sup>17</sup>We discard tokens that contain numerals, punctuation, or capital letters. As such,  $\tilde{V}$  includes more lexical items (with less noise and no subwords) than the original vocabulary  $V$ .

last two, i.e.  $\{3, 4, \dots, L - 2\}$ . We tune all hyper-parameters on the development split of SWORDS.

### 3.3 Results

Table 1 shows the results on SWORDS, along with baseline scores from previous work (with some bug fixes, as noted). The first row, **HUMANS**, indicates the agreement of two independent sets of annotators on (a subset of) SWORDS, and approximates the upper bound for this task. The second row, **CoInCo**, shows the accuracy of the gold standard substitutes in CoInCo, which are suggested by human annotators without access to substitution candidates<sup>18</sup> — this approximates how well humans perform when asked to elicit candidates themselves. The remainder of the rows above OURS denote baseline systems, all of which employ generative approaches. The first baseline uses **GPT-3** (Brown et al., 2020), and achieves the state-of-the-art in the strict setting. It generates substitutes based on “in-context learning”, where the model first reads several triplets of target sentences, queries, and gold-standard substitutes retrieved from the development set, and then performs on-the-fly inference on the test set. As such, it is not exactly comparable to the other fully unsupervised models. **BERT-K** generates substitutes based on Eqn. (5) by feeding the target sentence into BERT, and **BERT-M** works the same except that the target word is replaced by [MASK]. Both models further rerank the candidates based on Eqn. (4), using the last layer only; we show the performance without reranking as “w/o rerank” in Table 1. Yang et al. (2022) and Zhou et al. (2019) also use BERT to generate substitutes, and rerank them using their own method.

The rows below OURS indicate the performance of our approach using various off-the-shelf models. Our method with BERT substantially outperforms all the BERT-based baselines, even without the edit-distance heuristic (Section 2.2) or reranking method (Eqn. (4)). The best performing models are DeBERTa-V3, XLNet, and BART (Enc-Dec), all of which outperform the weakly-supervised GPT-3 model by a large margin in the strict setting; and XLNet even outperforms CoInCo in the lenient setting. The last three rows show the performance of the top-3 models when they are given the candidate words and rank them based on Eqn. (4), which emulates how the SWORDS annotators judged

<sup>18</sup>Since all of these words are in the substitute candidates of SWORDS, it cannot be evaluated under the strict setting.

| Models                               | S-07               | SW                 |
|--------------------------------------|--------------------|--------------------|
| HUMANS                               | —                  | 66.2               |
| Arefyev et al. (2020) (XLNet)        | 61.3 <sup>19</sup> | —                  |
| Michalopoulos et al. (2022) (LMs+WN) | 60.3               | —                  |
| Lacerra et al. (2021a) (BERT)        | 58.2               | —                  |
| Lacerra et al. (2021a) (BERT, sup)   | 60.5               | —                  |
| Zhou et al. (2019) (BERT)            | 60.5 <sup>20</sup> | 53.5 <sup>15</sup> |
| Lee et al. (2021) (BERT)             | 56.6               | 56.9               |
| OURS (Eqn. (4))                      |                    |                    |
| BERT                                 | 58.6               | 60.7               |
| mBERT                                | 45.4               | 52.0               |
| SpanBERT                             | 59.3               | 60.8               |
| MPNet                                | 61.5               | 59.5               |
| XLNet                                | 63.8               | <b>62.9</b>        |
| ELECTRA                              | 64.4               | 62.3               |
| DeBERTa-V3                           | <b>65.0</b>        | <b>62.9</b>        |
| BART (Enc)                           | 62.9               | 61.9               |
| BART (Dec)                           | 62.6               | 60.8               |
| BART (Enc-Dec)                       | 64.1               | 62.7               |
| SBERT (MPNet)                        | 61.0               | 62.5               |
| SimCSE (BERT)                        | 58.4               | 60.9               |
| NMT (mBART, Enc)                     | 46.0               | 51.5               |

Table 2: GAP scores on SemEval-07 and SWORDS. “LMs+WN” employs multiple language models and WordNet, and “BERT, sup” is a supervised model.

the words. The result shows that all the models still lag behind HUMANS, suggesting there is still substantial room for improvement. Interestingly, BART performs best when we average the scores obtained by its encoder and decoder, suggesting each layer captures complementary information. It is also intriguing to see that the discriminative models (DeBERTa-V3 and ELECTRA) perform much better than BERT, albeit they are not trained to generate words and not compatible with the previous generative approach. The sentence embedding models (SBERT, SimCSE) perform no better than the original models, which contrasts with their strong performance in sentence-level tasks. The multilingual models (mBERT, NMT) perform very poorly, even though the NMT model was fine-tuned on large English-X parallel corpora.

Table 2 shows the results for the ranking task on SemEval-07 and SWORDS. Michalopoulos et al. (2022) harness WordNet (Fellbaum, 1998) to obtain synsets of the target word and also their glosses, and employ BERT and RoBERTa (Liu et al., 2019)

<sup>19</sup>The original score reported by Arefyev et al. (2020) is 59.6, but we found we could improve this result by appending unscored OOV words to the ranked list in random order.

<sup>20</sup>Similar to Lacerra et al. (2021a) and Arefyev et al. (2020), we were unable to reproduce this score.

to rank candidates. The models proposed by [Lacerra et al. \(2021a\)](#) are different from the others in that they fine-tune BERT on lexical substitution data sets. They propose unsupervised (BERT) and supervised (BERT, sup) models, which are fine-tuned on automatically-generated or manually-annotated data. Table 2 shows that our method with BERT performs comparably with the unsupervised model of [Lacerra et al. \(2021a\)](#) without any fine-tuning, and outperforms [Zhou et al. \(2019\)](#) and [Lee et al. \(2021\)](#) (except for the score of [Zhou et al. \(2019\)](#) on SemEval-07, which couldn’t be reproduced in previous work). Just like the generation task, DeBERTa-V3 achieves the best performance on both data sets and establishes a new state-of-the-art. Other models also follow a similar trend to the generation results, e.g. BART performs best by combining its encoder and decoder, and the multilingual models perform very poorly. We hypothesise that their poor performance is mainly caused by suboptimal segmentation of English words. This hypothesis is also supported by the fact that DeBERTa-V3 has by far the largest vocabulary  $V$  of all models.<sup>21</sup>

### 3.4 Results on Italian Lexical Substitution

We further conduct an additional experiment on Italian, based on the data set from the EVALITA 2009 workshop ([Toral, 2009](#)). We report  $F$  scores given top-10 predictions as in the English generation task, plus two traditional metrics used in the workshop, namely *oot* and *best*, which compare the top-10 and top-1 predictions against the gold substitutes.<sup>22</sup> We lemmatise all the generated words to make them match the gold substitutes, following the SWORDS evaluation script.<sup>23</sup> We use the same hyper-parameters as for the English experiments, and Table 3 shows the results. [Hintz and Biemann \(2016\)](#) is a strong baseline that retrieves substitute candidates from MultiWordNet ([Pianta et al., 2002](#)) and ranks them using a supervised ranker model. We also implement BERT-K using an Italian BERT model ([Schweter, 2020](#)), with and without the reranking method. The results show that our approach substantially outperforms the baselines, confirming its effectiveness. However, our rerank-

<sup>21</sup>Note that  $V$  differs from  $\tilde{V}$ , the pre-defined vocabulary we used for all models. Appendix A compares the size of the model’s original vocabulary  $V$  across different models.

<sup>22</sup>We report precision only, as it is the same as recall under those metrics when predictions are made for every sentence.

<sup>23</sup>We used the Italian lemmatiser (it\_core\_news\_sm 3.2.0) in spaCy (ver. 3.2.2) ([Honnibal et al., 2020](#)).

|  | $F$         | best-P      | oot-P       |
|--|-------------|-------------|-------------|
| <a href="#">Hintz and Biemann (2016)</a> | —           | 16.2        | 41.3        |
| BERT-K                                   | 14.3        | 14.4        | 39.1        |
| (w/o rerank)                             | 15.6        | 17.4        | 43.3        |
| OURS (BERT)                              | 17.3        | 19.9        | 47.5        |
| (w/o rerank)                             | 17.5        | 19.1        | 48.4        |
| (w/o rerank, heuristic)                  | 17.5        | 17.4        | 48.7        |
| OURS (ELECTRA)                           | 19.0        | 21.0        | 51.2        |
| (w/o rerank)                             | 18.9        | <b>21.3</b> | 51.0        |
| (w/o rerank, heuristic)                  | <b>19.2</b> | 20.2        | <b>52.1</b> |

Table 3: The result of Italian lexical substitution.

|                 | BERT        | BART        | XLNet       | DeBERTa     |
|-----------------|-------------|-------------|-------------|-------------|
| $\lambda = 1$   | 34.1        | 26.2        | 32.9        | 35.8        |
| $\lambda = 0$   | 32.8        | 34.1        | 34.1        | 35.6        |
| $K = 1$         | 32.9        | 32.0        | 33.7        | 35.7        |
| $k$ is random   | 30.6        | 29.2        | 30.3        | 32.0        |
| w/o heuristic   | 32.4        | 32.0        | 33.4        | 35.7        |
| $\lambda = 0.7$ | 34.4        | 34.0        | 35.0        | 36.9        |
| + rerank        | <b>34.9</b> | <b>37.2</b> | <b>37.4</b> | <b>39.9</b> |

Table 4: Ablation studies of our method. The scores denote  $F_c$  in the strict setting on SWORDS.

ing method is not as effective as in English, which we attribute to the influence of grammatical gender in Italian (which we return to in Section 4.2). The heuristic improves best-P but harms  $F$  and oot-P, meaning it removes good candidates as well as bad ones, possibly because we used the threshold tuned on English.

## 4 Analysis

### 4.1 Ablation Studies

We perform ablation studies on SWORDS to see the effect of  $\lambda$  and the  $K$ -clustered embeddings, and also the heuristic based on edit distance. Table 4 shows the results. Overall, our method with  $\lambda = 0.7$  performs better than  $\lambda = 1$  or  $\lambda = 0$ , confirming the benefit of considering both in-context and out-of-context similarities. One interesting observation is that while BERT and DeBERTa perform better with  $\lambda = 1$  than with  $\lambda = 0$ , the opposite trend is observed for XLNet and especially BART (and hence the optimal value for  $\lambda$  is smaller than 0.7). This suggests that BART representations are highly influenced by context, containing much information that is not relevant to the semantics

|             | <i>a</i>    | <i>an</i>   | <i>un</i>   | <i>una</i>  | <i>la/le</i> | <i>il/i</i> |
|-------------|-------------|-------------|-------------|-------------|--------------|-------------|
| BERT-K      | 94.2        | 56.0        | 93.3        | 92.5        | 93.8         | 92.0        |
| (w/o r)     | 91.1        | 54.0        | 90.0        | 87.5        | 90.3         | 87.4        |
| OURS (BERT) | 88.4        | <u>24.0</u> | 81.9        | 89.2        | 92.6         | 85.2        |
| (w/o r)     | <u>86.8</u> | 18.0        | 70.5        | 66.7        | 69.7         | <u>64.6</u> |
| (w/o r, h)  | <u>86.8</u> | 44.0        | <u>62.4</u> | <u>65.0</u> | <u>69.4</u>  | 60.7        |
| Gold        | 86.8        | 26.9        | 55.6        | 63.3        | 69.0         | 67.0        |

Table 5: The percentage of generated and gold substitutes whose initial sound or grammatical gender agrees with the corresponding English or Italian articles. “r” and “h” denote rerank and heuristic, respectively. The closest numbers to Gold are underlined.

of the target word; we further confirm this in the next section. When we set the cluster size  $K$  to 1, the performance of all the models drops sharply, indicating the effectiveness of the clustered embeddings. When we retrieve the cluster of  $y$  at random instead of the closest one to  $x$  in Eqn. (2), the performance decreases substantially, suggesting each cluster captures different semantics. The heuristic consistently improves the performance, filtering out derivationally-related yet semantically-dissimilar words to the target word. Lastly, our reranking method substantially improves the performance of all the models, demonstrating that it is important to incorporate the target context  $c$  into both the target and candidate word representations.

## 4.2 Effects of Morphosyntactic Agreement

Compared to previous generative approaches, our method does not depend on the generation probabilities of language models, and hence we expect it to be less sensitive to morphosyntactic agreement effects. To investigate this, we analyse the performance on noun target words which immediately follow one of the following articles: *a* or *an* in English, and *una*, *la*, *le*, *un*, *il* or *i* in Italian. The first three Italian articles are used with feminine nouns, and the rest with masculine ones. Our hypothesis is that generative methods will be highly biased by these articles, despite the gold standard being *semantically* annotated, and thus largely oblivious to local morphosyntactic agreement effects.

Table 5 shows the percentage of top-10 predicted candidates that agree with the article.<sup>24</sup> It demon-

<sup>24</sup>We retrieve Italian gender information using a dictionary API ([https://github.com/sphoneix22/italian\\_dictionary](https://github.com/sphoneix22/italian_dictionary)), and English phonetic information using CMUdict (<https://github.com/cmuspinx/cmudict>) accessed via

| Models         | <i>a</i> | <i>an</i> |
|----------------|----------|-----------|
| BERT-K         | 94.2     | 56.0      |
| BERT-M         | 99.5     | 88.0      |
| BERT           | 88.4     | 24.0      |
| SpanBERT       | 97.9     | 46.0      |
| MPNet          | 91.6     | 52.0      |
| XLNet          | 83.2     | 38.0      |
| BART (Enc)     | 89.5     | 62.0      |
| BART (Dec)     | 91.1     | 46.0      |
| BART (Enc-Dec) | 89.5     | 58.0      |
| DeBERTa-V3     | 86.8     | 32.0      |
| ELECTRA        | 88.4     | 36.0      |

Table 6: The percentage of substitutes whose initial sound agrees with the corresponding English articles.

strates that the prediction of BERT-K is highly affected by the preceding article as expected, resulting in substitutes which don’t satisfy this constraint being assigned low probabilities. In contrast, the results of our method are more balanced and close to the gold standard.<sup>25</sup> Conversely, our reranking method actually *increases* the bias greatly, suggesting that the contextualised embeddings  $f^\ell(x, c)$  and  $f^\ell(y, c)$  in Eqn. (4) become similar when  $x$  and  $y$  collocate similarly with the words in the context  $c$  — overall, this leads to better results, but actually hurts in cases of local agreement effects biasing the results. This is one reason why reranking was not as effective in Italian as in English, as agreement effects are stronger in Italian.

Table 6 shows the result when we use different pre-trained models in English. First, it shows that BERT-M is more sensitive to the articles than BERT-K, indicating the strong morphophonetic agreement effect on the masked word prediction. Among the pre-trained language models used by our method, SpanBERT and BART are the most sensitive to the article *a* and *an*, respectively. This suggests that the embeddings  $f(x, c)$  obtained from these models are highly sensitive to the context  $c$ , partly explaining why BART performs very poorly with  $\lambda = 1$ , as shown in Section 4.1. Lastly, Table 7 shows examples of predicted substitutes when the article *an* comes before the target word. It shows that BERT-K and OURS with BART tend to retrieve words that start with a vowel sound, as quan-

NLTK (Steven et al., 2009).

<sup>25</sup>Note that the big jump in results for *an* is based on a small number of instances (5 sentences).



|                    |  |
|--------------------|--|
| Context            | The loan may be extended by the McAlpine group for an additional year with an <b>increase</b> in the conversion price to \$2.50 a share.   |
| Gold (Conceivable) | <i>boost, gain, raise, hike, rise, swell, surge, upsurge, enlargement, growth, addition, escalation, expansion, upgrade, cumulation, swelling, exaggeration, step-up</i>                           |
| BERT-K             | increased, <b>rise</b> , enhancement, increasing, <b>addition</b>  |
| OURS (BERT)        | <b>rise, raise</b> , change, reduce, reduction   |
| OURS (BART)        | <b>rise</b> , uptick, <b>hike</b> , improvement, upping  |
| OURS (DeBERTa-V3)  | <b>boost, rise, raise, hike</b> , reduction  |
| Context            | Under an <b>accord</b> signed yesterday, the government and Union Bank of Finland would become major shareholders in the new company, each injecting 100 million Finnish markkaa (\$23.5 million). |
| Gold (Conceivable) | <i>arrangement, agreement, pact, contract, deal, treaty</i>  |
| BERT-K             | understanding, <b>agreement, pact, arrangement</b> , agreeing  |
| OURS (BERT)        | <b>agreement, pact, treaty</b> , understanding, <b>deal</b>  |
| OURS (BART)        | <b>agreement, deal, pact, arrangement, treaty</b>  |
| OURS (DeBERTa-V3)  | <b>pact, agreement, deal, arrangement, treaty</b>  |

Table 7: Top-5 predictions when the article *an* comes before the target word (*increase* or *accord*). Gold shows a list of “conceivable” words sorted by their annotated scores (with “acceptable” words shown in italic, and multiword expressions omitted from the table). Words included in Gold are boldfaced.

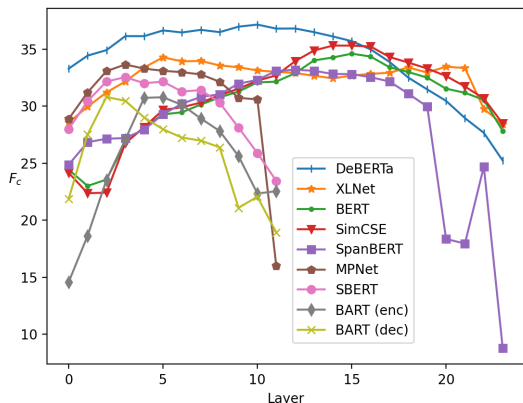


Figure 1: Layer-wise performance ( $F_c$ ) on SWORDS.

tatively described in Table 6.

### 4.3 Layer-Wise Performance

We analyse the performance on SWORDS using different layers in Figure 1 (w/o rerank).<sup>26</sup> First, we clearly see that middle layers perform better than the first or last ones, for all models.<sup>27</sup> The performance of BERT peaks at layer 16, in contrast with previous findings that the first quarter of layers perform best on context-independent word similarity tasks (Bommasani et al., 2020), likely

<sup>26</sup>We perform qualitative analysis in Appendix D.

<sup>27</sup>We see a similar trend in the ranking task (Appendix B).

because lexical substitution critically relies on context.<sup>28</sup> Our method using multiple layers performs mostly as well as using the best layer without the need to perform model-wise layer selection (see Table 11 in Appendix C). The last layer performs very poorly for all models, highlighting the limitation of the previous approach which uses the last layer only (Eqn. (5)). The downward trend is particularly evident for MPNet, BART (dec), and SpanBERT; for BART and SpanBERT, we attribute this to the fact that their last-layer representations of the word at position  $t$  are used to predict the next word  $w_{t+1}$ , or  $u$  neighbouring words  $\{w_{t-u}, \dots, w_{t-1}\}$  or  $\{w_{t+1}, \dots, w_{t+u}\}$ .<sup>29</sup> This training objective may also lead to their sensitivity to articles before the target word, as shown in Section 4.2. Interestingly, the sentence-embedding models (SBERT and SimCSE) are no exception to the downward trend, which is somewhat counter-intuitive given that their last layer representations are fine-tuned (and used during inference) to perform semantic downstream tasks. Importantly, they do not perform better than the original models (MPNet and BERT), although in the ranking task, both models benefit moderately from fine-tuning (see Appendix B).

<sup>28</sup>In fact, Tenney et al. (2019) show that high-level semantic information is encoded in higher layers.

<sup>29</sup>SpanBERT does this for Span Boundary Objective.

|               | # Matched Words |            |             | $F_c$       |
|---------------|-----------------|------------|-------------|-------------|
|               | low             | med        | high        |             |
| BERT-K        | 121             | 144        | 2002        | 30.4        |
| BERT (5k)     | 31              | 52         | 2017        | 28.1        |
| BERT (10k)    | 72              | 111        | 2249        | 32.6        |
| BERT (20k)    | 164             | 220        | 2194        | 34.5        |
| BERT (30k)    | 241             | 267        | 2095        | 34.9        |
| BART (30k)    | 380             | 274        | 2123        | 37.2        |
| XLNet (30k)   | 395             | <b>292</b> | 2106        | 37.4        |
| DeBERTa (30k) | <b>429</b>      | 287        | <b>2262</b> | <b>39.9</b> |

Table 8: The number of correctly predicted substitutes, grouped by their frequency in monolingual data. The number in brackets shows the size of the vocabulary  $\tilde{V}$ .

| $K$   | 1    | 2    | 4    | 8           | 16          |
|-------|------|------|------|-------------|-------------|
| $F_a$ | 22.8 | 23.1 | 23.2 | 23.3        | <b>23.4</b> |
| $F_c$ | 36.0 | 36.7 | 36.7 | <b>36.9</b> | 36.7        |

Table 9: Results with different numbers of clusters.

#### 4.4 Analysis of Word Frequency

One of the strengths of our approach is that it can generate low-frequency substitutes that are OOV words in the original vocabulary. To confirm this, we analyse how well our method can generate low-frequency words from different vocabulary sizes  $\tilde{V}$ . Table 8 shows the results, in which we experiment with our BERT-based model with the vocabulary sizes of 5k, 10k, 20k, and 30k. The columns under “# Matched Words” show the numbers of correctly-predicted words, grouped by frequency range: low, med, and high denote words with frequency  $<50k$ ,  $50k-100k$ , and  $>100k$  in a large web corpus. The table shows that our method with 30k words generates nearly twice as many low-frequency substitutes as the baseline. Our method with 10k words still outperforms BERT-K in  $F_c$ , demonstrating its effectiveness. The last three rows show the performance of our method using other models, further demonstrating its ability to predict low-frequency words.

#### 4.5 Effects of Cluster Size

Finally, we analyse the effect of the cluster size  $K$  for ELECTRA, as shown in Table 9. While a larger cluster size yields better performance, the improvement is marginal. Rather than using a fixed

$K$ , in future work we are interested in dynamically selecting the number of clusters per word.

## 5 Related Work

In the pre-BERT era, most lexical substitution methods employed linguistic resources such as WordNet (Fellbaum, 1998) to obtain substitute candidates (Szarvas et al., 2013; Hintz and Biemann, 2016). However, recent studies have shown that pre-trained language models such as BERT outperform these models without any external lexical resources. For instance, Zhou et al. (2019) feed a target sentence into BERT while partially masking the target word using dropout (Srivastava et al., 2014), and generate substitutes based on the probability distribution at the target word position. The masking strategy was shown to be effective on SemEval-07 but not on SWORDS. Similarly, Yang et al. (2022) feed two sentences into BERT, concatenating the target sentence with itself but with the target word replaced by [MASK], and predict words based on the mask-filling probability. Michalopoulos et al. (2022) augment pre-trained language models with WordNet and outperform Zhou et al. (2019). Lacerra et al. (2021a) fine-tune BERT on lexical substitution data sets that are automatically generated using BERT. They show that this approach is effective at ranking, and that adding manually-annotated data further boosts performance. Lacerra et al. (2021b) fine-tune BART on human-annotated data, and make it generate a list of substitutes given a target sentence in an end-to-end manner. They show that this generative approach rivals Zhou et al. (2019). Note that all of these recent models are evaluated on English only.

## 6 Conclusion

We present a new unsupervised approach to lexical substitution using pre-trained language models. We showed that our method substantially outperforms previous methods on English and Italian data sets, establishing a new state-of-the-art. By comparing performance on lexical substitution using different layers, we found that middle layers perform better than first or last layers. We also compared the substitutes predicted by the previous generative approach and our method, and showed that our approach works better at predicting low-frequency substitutes and reduces morphophonetic or morphosyntactic biases induced by article-noun agreement in English and Italian.

## References

- 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.
- David Arthur and Sergei Vassilvitskii. 2007. [K-means++: The advantages of careful seeding](#). In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07*, page 1027–1035, USA. Society for Industrial and Applied Mathematics.
- Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. [Interpreting pretrained contextualized representations via reductions to static embeddings](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Gabriella Chronis and Katrin Erk. 2020. [When is a bishop not like a rook? When it’s like a rabbi! Multi-prototype BERT embeddings for estimating semantic relationships](#). In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 227–244, Online. Association for Computational Linguistics.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [ELECTRA: Pre-training text encoders as discriminators rather than generators](#). In *International Conference on Learning Representations*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [SimCSE: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Patrick Hanks. 2012. [The Corpus Revolution in Lexicography](#). *International Journal of Lexicography*, 25(4):398–436.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. [DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing](#). *CoRR*, abs/2111.09543.
- 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.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spaCy: Industrial-strength natural language processing in Python](#).
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. [SpanBERT: Improving pre-training by representing and predicting spans](#). *Transactions of the Association for Computational Linguistics*, 8:64–77.
- Kazuaki Kishida. 2005. [Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments](#). *NII Technical Report*.
- 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, and Roberto Navigli. 2021a. [ALaSca: an automated approach for large-scale lexical substitution](#). In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence.
- 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.

- 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 *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4362–4379, Online. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. 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](#). *CoRR*, abs/1907.11692.
- S. Lloyd. 1982. [Least squares quantization in pcm](#). *IEEE Transactions on Information Theory*, 28(2):129–137.
- Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval '07*, page 48–53, USA. Association for Computational Linguistics.
- Oren Melamud, Omer Levy, and Ido Dagan. 2015. [A simple word embedding model for lexical substitution](#). In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, Denver, Colorado. Association for Computational Linguistics.
- George 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.
- Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. [A monolingual approach to contextualized word embeddings for mid-resource languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.
- Emanuele Pianta, Luisa Bentivogli, and Christian Giardi. 2002. MultiWordNet: Developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*, pages 293–302, Mysore, India.
- 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*. Association for Computational Linguistics.
- Stefan Schweter. 2020. [Italian BERT and ELECTRA models](#). <https://doi.org/10.5281/zenodo.4263142>.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. [MPNet: Masked and permuted pre-training for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. [Dropout: A simple way to prevent neural networks from overfitting](#). *Journal of Machine Learning Research*, 15:1929–1958.
- Bird Steven, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O’Reilly Media Inc.
- György Szarvas, Chris Biemann, and Iryna Gurevych. 2013. [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.
- Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. [Multilingual translation from denoising pre-training](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [BERT rediscovers the classical NLP pipeline](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
- Antonio Toral. 2009. [The lexical substitution task at EVALITA 2009](#). In *EVALITA workshop, 11th Congress of Italian Association for Artificial Intelligence*, Reggio Emilia, Italy.
- 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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT make any sense? interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 161–170, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. *Transformers: State-of-the-art natural language processing*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. 2022. *Tracing text provenance via context-aware lexical substitution*. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, Virtual.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. *XLNet: Generalized autoregressive pretraining for language understanding*. In *Advances in Neural Information Processing Systems*, volume 32.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. *BERT-based lexical substitution*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy. Association for Computational Linguistics.

## A Details of Pre-trained Models

Table 10 describe the details of the pre-trained models used in our experiments. We sourced these models from the Transformers library (Wolf et al., 2020) except for SpanBERT, which we obtained from the original GitHub repository (<https://github.com/facebookresearch/SpanBERT>).

## B Layer-Wise Ranking Performance

Figure 2 shows the layer-wise performance in the ranking task. Similar to the generation results (Figure 1), middle layers perform better than the first or last layers. It also shows that sentence-embedding models (SimCSE/SBERT) outperform their original models (BERT/MPNet) for several layers, different from the generation results where they perform similarly. This suggests that fine-tuning on semantic downstream tasks improves the capacity

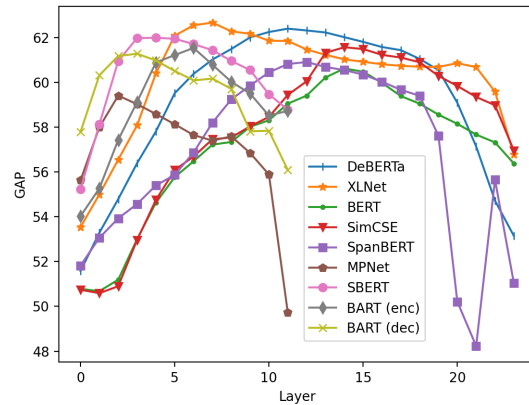


Figure 2: Layer-wise performance (GAP) on SWORDS.

of the model to differentiate subtle semantic differences between synonymous words, but not their ability to retrieve relevant words from a large pool of words; it also suggests that optimal representations for these objectives might differ.

## C Effectiveness of Using Multiple Layers

Table 11 shows the generation and ranking performance of our model on SWORDS using different layers. It shows that our method using multiple layers  $\ell \in Z$  performs comparably or even better than selecting the best layer tuned on the *test* set for each model. It also shows that the best-performing layer differs across models, suggesting they capture lexical information in a different manner.

## D Examples of Generated Substitutes

Table 12 shows examples of substitutes generated by our method using different layers (without reranking). It shows that the words retrieved by each layer are very different, indicating that each layer encodes very different information about the input word. For instance, given the target word *care*, the first layer of BERT and BART-Enc/Dec retrieves a large number of words that contain the target word as a sub-morpheme (e.g. *aftercare*, *carefree*).<sup>30</sup> This is presumably because the first-layer representations are highly affected by the input word embedding, and hence result in retrieving words that share the same subword token (e.g. *care ##free*) regardless of the semantic similarity. The last layer also performs poorly (as previously shown in Figure 1), e.g. BART-DEC (L12) retrieves

<sup>30</sup>Since the edit distances between these words and the target word *care* are not greater than the threshold (0.5), they weren’t filtered out by our heuristic.

| Models            | # Layer | Emb Size | $ V $   | Model Path   |
|-------------------|---------|----------|---------|--|
| BERT              | 24      | 1024     | 30522   | bert-large-uncased                                 |
| mBERT             | 12      | 768      | 105879* | bert-base-multilingual-uncased                     |
| SpanBERT          | 24      | 1024     | 30522   | spanbert-large-cased                               |
| MPNet             | 12      | 768      | 30527   | microsoft/mpnet-base                               |
| XLNet             | 24      | 1024     | 32000   | xlnet-large-cased                                  |
| ELECTRA           | 24      | 1024     | 30522   | google/electra-large-discriminator                 |
| DeBERTa-V3        | 24      | 1024     | 128000  | microsoft/deberta-v3-large                         |
| BART (Enc/Dec)    | 12      | 1024     | 50265   | facebook/bart-large                                |
| SBERT (MPNet)     | 12      | 768      | 30527   | sentence-transformers/all-mpnet-base-v2            |
| SimCSE (BERT)     | 24      | 1024     | 30522   | princeton-nlp/sup-simcse-bert-large-uncased        |
| NMT (mBART, Enc)  | 12      | 1024     | 250054* | facebook/mbart-large-50-one-to-many-mmt            |
| BERT (Italian)    | 12      | 768      | 31102   | dbmdz/bert-base-italian-xxl-uncased                |
| ELECTRA (Italian) | 12      | 768      | 31102   | dbmdz/electra-base-italian-xxl-cased-discriminator |

Table 10: Details of the pre-trained models used in this paper.  $|V|$  denotes the original vocabulary size of each model. \*The vocabularies of mBERT and mBART contain a great number of non-English words.

| Layer      | Generation Performance ( $F_c$ ) |             |      |                  |              | Ranking Performance (GAP) |        |      |                  |              |
|------------|----------------------------------|-------------|------|------------------|--------------|---------------------------|--------|------|------------------|--------------|
|            | First                            | Middle      | Last | Best             | $\ell \in Z$ | First                     | Middle | Last | Best             | $\ell \in Z$ |
| BERT       | 24.4                             | 32.2        | 27.8 | <b>34.6</b> (16) | 34.4         | 50.8                      | 59.1   | 56.4 | 60.6 (15)        | <b>60.7</b>  |
| SpanBERT   | 24.9                             | 33.1        | 8.8  | <b>33.2</b> (13) | 31.1         | 51.8                      | 60.8   | 51.0 | <b>60.9</b> (13) | 60.8         |
| MPNet      | 28.9                             | 33.1        | 16.0 | 33.7 (4)         | <b>33.8</b>  | 55.6                      | 58.1   | 49.7 | 59.4 (3)         | <b>59.5</b>  |
| XLNet      | 28.6                             | 33.0        | 28.4 | 34.3 (6)         | <b>35.0</b>  | 53.5                      | 61.9   | 56.8 | 62.7 (8)         | <b>62.9</b>  |
| DeBERTa-V3 | 33.3                             | 36.8        | 25.2 | <b>37.2</b> (11) | 36.9         | 51.6                      | 62.4   | 53.2 | 62.4 (12)        | <b>62.9</b>  |
| BART (Enc) | 14.6                             | <b>30.8</b> | 22.6 | <b>30.8</b> (6)  | 30.0         | 54.0                      | 61.2   | 58.7 | 61.5 (7)         | <b>61.9</b>  |
| BART (Dec) | 21.9                             | 28.0        | 18.9 | <b>30.8</b> (3)  | 29.0         | 57.8                      | 60.5   | 56.1 | <b>61.3</b> (4)  | 60.8         |

Table 11: Generation and ranking performance of our approach on SWORDS using the first, middle ( $\frac{L}{2}$ th), last ( $L$ th), or best layer tuned on the *test* set (the corresponding layer denoted in brackets); or using multiple layers  $\ell \in Z: \{3, 4, \dots, L-2\}$  ( $L = 12$  for MPNet and BART, and 24 otherwise). Generation and ranking performance across all layers is illustrated in Figure 1 and Figure 2.

*participation* as the closest word to the target word *interest*. This is because the last-layer representations of BART-decoder are used to directly predict the next word *in* after *interest* in the target sentence, and in fact, most of the retrieved words (e.g. *uptick*, *faith*, *surge*) are those that often collocate with *in*. Oddly, BART-Enc predicts a large number of substitutes that consist of multiple words (segmented by the tokeniser), none of which are relevant to the target word, e.g. *aswell*, *todo*, and *inbetween* as substitutes for *interest*. In fact, the number of such words increases (and the performance decreases) as the hyper-parameter  $\lambda$  gets bigger (which increases the influence of  $f(x, c)$  on the predictions). One possible interpretation is that the last layer representations of the BART encoder may contain vague contextual information rather than the lexi-

cal information of the input word, since they are used by the decoder to predict various words (esp. masked words) during pre-training. Lastly, another interesting observation is that, for the target word *interest*, the last layer representations of BERT and BART-enc retrieve a lot of words that start from a vowel sound, despite the absence of the article *an* before *interest*, suggesting that the embeddings contain some morphophonetic information.

|                    |  |
|--------------------|--|
| Context            | I say I do not <b>care</b> about law, I care about service and she should care about money.  |
| Gold (Conceivable) | <i>worry</i> , think, mind, desire, love, tend, cherish, consider, stress, concern, bother, watch  |
| BERT (L3-22)       | matter, caregiving, carefree, <b>worry</b> , <b>concern</b> , know, pay, caregivers, <b>love</b> , like  |
| BART (L3-10)       | <b>concern</b> , matter, carelessness, caretaker, carelessly, carefree, caretakers, <b>worry</b> , <b>bother</b> , <b>mind</b>   |
| BERT (L1)          | caregiving, carefree, caregiver, caregivers, aftercare, childcare, healthcare, skincare, custody, affections   |
| BERT (L12)         | matter, caregiving, <b>worry</b> , carefree, fret, <b>love</b> , despise, loathe, resent, pay  |
| BERT (L24)         | matter, <b>worry</b> , pay, <b>concern</b> , know, look, give, take, <b>bother</b> , <b>think</b>  |
| BART-Enc (L1)      | aftercare, caregiving, caretaker, carefree, carelessly, carelessness, caretakers, healthcare, caregivers, medicare   |
| BART-Enc (L6)      | <b>concern</b> , todo, caretaker, interest, carelessness, careless, disinterested, disdain, pay, carelessly  |
| BART-Enc (L12)     | todo, aswell, beleive, inbetween, zealand, pay, <b>concern</b> , usefull, noone, ofcourse  |
| BART-Dec (L1)      | caregiving, carelessness, caretaker, carelessly, carefree, aftercare, caretakers, <b>concern</b> , healthcare, <b>worry</b>  |
| BART-Dec (L6)      | <b>concern</b> , <b>bother</b> , <b>worry</b> , commit, reckon, dispose, shit, grieve, pay, strive   |
| BART-Dec (L12)     | <b>worry</b> , damn, <b>concern</b> , inquire, complain, passionate, shit, <b>think</b> , talk, whine  |
| Context            | “I’m starting to see more business transactions,” says Andrea West of American Telephone & Telegraph Co., noting growing <b>interest</b> in use of 900 service for stock sales, software tutorials and even service contracts. |
| Gold (Conceivable) | <i>interestedness</i> , enthusiasm, demand, attraction, popularity, excitement, curiosity, activity, importance, notice, significance, involvement, relevance, note, gain, passion, influence, accrual, concernment            |
| BERT (L3-22)       | <b>curiosity</b> , <b>enthusiasm</b> , intrigued, desire, concern, fascination, <b>passion</b> , attention, <b>excitement</b> , fondness   |
| BART (L3-10)       | fascination, <b>enthusiasm</b> , appetite, <b>curiosity</b> , <b>excitement</b> , concern, inclination, eagerness, desire, <b>involvement</b>  |
| BERT (L1)          | concern, <b>importance</b> , <b>curiosity</b> , investment, <b>involvement</b> , attention, focussed, fascination, focus, <b>significance</b>  |
| BERT (L12)         | <b>curiosity</b> , <b>enthusiasm</b> , concern, fascination, confidence, unease, belief, <b>excitement</b> , desire, <b>passion</b>  |
| BERT (L24)         | appetite, attracting, attractiveness, actively, <b>demand</b> , attention, intrigued, <b>popularity</b> , <b>enthusiasm</b> , flocking   |
| BART-Enc (L1)      | fascination, concern, intrigue, <b>involvement</b> , <b>relevance</b> , investment, stake, <b>curiosity</b> , <b>enthusiasm</b> , trustworthiness  |
| BART-Enc (L6)      | <b>enthusiasm</b> , fascination, appetite, <b>curiosity</b> , intrigued, <b>excitement</b> , intrigue, inclination, eagerness, enjoyment   |
| BART-Enc (L12)     | todo, aswell, inbetween, <b>enthusiasm</b> , fascination, eagerness, appetite, intrigued, attention, inclination   |
| BART-Dec (L1)      | fascination, <b>involvement</b> , intrigue, concern, participation, <b>curiosity</b> , investment, <b>excitement</b> , intrigued, <b>importance</b>  |
| BART-Dec (L6)      | fascination, appetite, <b>involvement</b> , engagement, affinity, uptake, <b>demand</b> , inclination, participation, appreciation   |
| BART-Dec (L12)     | participation, uptick, delight, decline, increase, spike, faith, decrease, grounding, surge  |

Table 12: Examples of substitutes predicted by our method (w/o rerank) using different layers. Gold shows a list of “conceivable” words sorted by their annotated scores (with “acceptable” words shown in italic, and multiword expressions omitted from the table). Words included in Gold are boldfaced.