Always Keep your Target in Mind: Studying Semantics and Improving Performance of Neural Lexical Substitution

Nikolay Arefyev[1,2,3], **Boris Sheludko**[1,2], **Alexander Podolskiy**[1*], and **Alexander Panchenko**[4]

[1]Samsung Research Center Russia, Moscow, Russia
[2]Lomonosov Moscow State University, Moscow, Russia
[3]HSE University, Moscow, Russia
[4]Skolkovo Institute of Science and Technology, Moscow, Russia
narefjev@cs.msu.ru
{b.sheludko,a.podolskiy}@samsung.com
a.panchenko@skoltech.ru

Abstract

Lexical substitution, i.e. generation of plausible words that can replace a particular target word in a given context, is an extremely powerful technology that can be used as a backbone of various NLP applications, including word sense induction and disambiguation, lexical relation extraction, data augmentation, etc. In this paper, we present a large-scale comparative study of lexical substitution methods employing both rather old and most recent language and masked language models (LMs and MLMs), such as context2vec, ELMo, BERT, RoBERTa, XLNet. We show that already competitive results achieved by SOTA LMs/MLMs can be further substantially improved if information about the target word is injected properly. Several existing and new target word injection methods are compared for each LM/MLM using both intrinsic evaluation on lexical substitution datasets and extrinsic evaluation on word sense induction (WSI) datasets. On two WSI datasets we obtain new SOTA results. Besides, we analyze the types of semantic relations between target words and their substitutes generated by different models or given by annotators.

1 Introduction

Lexical substitution is the task of generating words that can replace a given word in a given textual context. For instance, in the sentence "*My daughter purchased a new car*" the word *car* can be substituted by its synonym *automobile*, but also with co-hyponym *bike*, or even hypernym *motor vehicle* while keeping the original sentence grammatical. Lexical substitution has been proven effective in various applications, such as word sense induction (Amrami and Goldberg, 2018), lexical relation extraction (Schick and Schütze, 2020), paraphrase generation, text simplification, textual data augmentation, etc. Note that the preferable type (e.g., synonym, hypernym, co-hyponym, etc.) of generated substitutes depends on the task at hand.

The new generation of language and masked language models (LMs/MLMs) based on deep neural networks enabled a profound breakthrough in almost all NLP tasks. These models are commonly used to perform pre-training of deep neural networks, which are then fine-tuned to some final task different from language modeling. However, in this paper we study how the progress in unsupervised pre-training over the last five years affected the quality of lexical substitution. We adapt context2vec (Melamud et al., 2016), ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019) to solve lexical substitution task without any fine-tuning, but using additional techniques to ensure similarity of substitutes to the target word, which we call target word injection techniques. We provide the first large-scale comparison of various neural LMs/MLMs with several target word injection methods on lexical substitution and WSI tasks. Our research questions are the following (i) which pre-trained models are the best for substitution in context, (ii) additionally to pre-training larger

*Left Samsung

models on more data, what are the other ways to improve lexical substitution, and (iii) what are the generated substitutes semantically. More specifically, the main contributions of the paper are as follows[1]:

- A comparative study of five neural LMs/MLMs applied for lexical substitution based on both intrinsic and extrinsic evaluation.
- A study of methods of target word injection for further lexical substitution quality improvement.
- An analysis of types of semantic relations (synonyms, hypernyms, co-hyponyms, etc.) produced by neural substitution models as well as human annotators.

2 Related Work

Solving the lexical substitution task requires finding words that are both appropriate in the given context and related to the target word in some sense (which may vary depending on the application of generated substitutes). To achieve this, unsupervised substitution models heavily rely on distributional similarity models of words (DSMs) and language models (LMs). Probably, the most commonly used DSM is *word2vec* model (Mikolov et al., 2013). It learns word embeddings and context embeddings to be similar when they tend to occur together, resulting in similar embeddings for distributionally similar words. Contexts are either nearby words or syntactically related words (Levy and Goldberg, 2014). In (Melamud et al., 2015b) several metrics for lexical substitution were proposed based on embedding similarity of substitutes both to the target word and to the words in the given context. Later (Roller and Erk, 2016) improved this approach by switching to dot-product instead of cosine similarity and applying an additional trainable transformation to context word embeddings.

A more sophisticated *context2vec* model producing embeddings for a word in a particular context (contextualized word embeddings) was proposed in (Melamud et al., 2016) and was shown to outperform previous models in a ranking scenario when candidate substitutes are given. The training objective is similar to word2vec, but context representation is produced by two LSTMs (a forward and a backward for the left and the right context), in which final outputs are combined by feed-forward layers. For lexical substitution, candidate word embeddings are ranked by their similarity to the given context representation. A similar architecture consisting of a forward and a backward LSTM is employed in ELMo (Peters et al., 2018). However, in ELMo each LSTM was trained with the LM objective instead. To rank candidate substitutes using ELMo (Soler et al., 2019) proposed calculating cosine similarity between contextualized ELMo embeddings of the target word and all candidate substitutes. This requires feeding the original example with the target word replaced by one of the candidate substitutes at a time. The average of outputs at the target timestep from all ELMo layers performed best. However, they found context2vec performing even better and explained this by the negative sampling training objective, which is more related to the task.

Recently, Transformer-based models pre-trained on huge corpora with LM or similar objectives have shown SOTA results in various NLP tasks. BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) were trained to restore a word replaced with a special `[MASK]` token given its full left and right contexts (masked LM objective), while XLNet (Yang et al., 2019) predicted a word at a specified position given only some randomly selected words from its context (permutation LM objective). In (Zhou et al., 2019), BERT was reported to perform poorly for lexical substitution (which is contrary to our experiments), and two improvements were proposed to achieve SOTA results using it. Firstly, dropout is applied to the target word embedding before showing it to the model. Secondly, the similarity between the original contextualized representations of the context words and their representations after replacing the target by one of the possible substitutes are integrated into the ranking metric to ensure minimal changes in the sentence meaning. This approach is very computationally expensive, requiring calculation of several forward passes of BERT for each input example, depending on the number of possible substitutes. We are not aware of any work applying XLNet for lexical substitution, but our experiments show that it outperforms BERT by a large margin.

[1]The repository for this paper: `https://github.com/bsheludko/lexical-substitution`

Supervised approaches to lexical substitution include (Szarvas et al., 2013a; Szarvas et al., 2013b; Hintz and Biemann, 2016). These approaches rely on manually curated lexical resources like WordNet, so they are not easily transferable to different languages, unlike those described above. Also, the latest unsupervised methods were shown to perform better (Zhou et al., 2019).

3 Neural Language Models for Lexical Substitution with Target Word Injection

To generate substitutes, we introduce several substitute probability estimators, which are models taking a text fragment and a target word position in it as input and producing a list of substitutes with their probabilities. To build our substitute probability estimators we employ the following LMs/MLMs: context2vec (Melamud et al., 2016), ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019). These models were selected to represent the progress in unsupervised pre-training with language modeling and similar tasks over the last five years. Given a target word occurrence, the basic approach for models like context2vec and ELMo is to encode its context and predict the probability distribution over possible center words in this particular context. This way, the model does not see the target word. For MLMs, the same result can be achieved by masking the target word. This basic approach employs the core ability of LMs/MLMs of predicting words that fit a particular context. However, these words are often not related to the target. The information about the target word can improve generated substitutes, but what is the best method of injecting this information is an open question.

3.1 Target Word Injection Methods

We experiment with several methods to introduce information about the original target word into neural lexical substitution models and show that their performance differs significantly. Suppose we have an example LTR, where T is the target word, and $C = (L, R)$ is its context (left and right correspondingly). For instance, the occurrence of the target word *fly* in the sentence "*Let me fly away!*" will be represented as $T =$*"fly"*, $L =$*"Let me"*, $R =$*"away!"*.

+embs This method combines a distribution provided by a context-based substitute probability estimator $P(s|C)$ with a distribution based on the proximity of possible substitutes to the target $P(s|T)$. The proximity is computed as the inner product between the respective embeddings, and the softmax function is applied to get a probability distribution. However, if we simply multiply these distributions, the second will have almost no effect because the first is very peaky. To align the orders of distributions, we use temperature softmax with carefully selected temperature hyperparameter: $P(s|T) \propto \exp(\frac{\langle emb_s, emb_T \rangle}{\mathcal{T}})$. The final distribution is obtained by the formula $P(s|C,T) \propto \frac{P(s|C)P(s|T)}{P(s)^\beta}$. For $\beta = 1$, this formula can be derived by applying the Bayes rule and assuming conditional independence of C and T given s. Other values of β can be used to penalize frequent words, more or less. Our current methods are limited to generating only substitutes from the vocabulary of the underlying LM/MLM. Thus, we take word or subword embeddings of the same model we apply the injection to. Other word embeddings like word2vec may perform better, but we leave these experiments for future work.

Word probabilities $P(s)$ are retrieved from wordfreq library[2] for all models except ELMo. Following (Arefyev et al., 2019), for ELMo, we calculate word probabilities from word ranks in the ELMo vocabulary (which is ordered by word frequencies) based on Zipf-Mandelbrot distribution and found it performing better presumably due to better correspondence to the corpus ELMo was pre-trained on.

Dynamic patterns Following the approach proposed in (Amrami and Goldberg, 2018), we replace the target word T by "T and _" (e.g. *"Let me fly and _ away!"*). Then some LM/MLM is employed to predict possible words at timestep "_". Thus, dynamic patterns provide information about the target word to the model via Hearst-like patterns.

Duplicate input This method duplicates the original example while hiding the target word (e.g., *"Let me fly away! Let me _ away!"*). Then possible words at timestep "_" are predicted. It is based on our

[2] `https://pypi.org/project/wordfreq`

observation that Transformer-based MLMs are very good at predicting words from the context when they fit the specified timestep (copying) while still giving a high probability to their distributionally similar alternatives.

Original input For MLMs such as BERT and RoBERTa, instead of masking the target word, we can leave it intact. Thus, the model predicts possible words at the target position while receiving the original target at its input. We noticed that unlike duplicate input, in this case, the MLM often puts the whole probability mass to the original target and gives very small probabilities to other words making their ranking less reliable. For XLNet, we can use such attention masks that the context words can see the target word in the content stream. Thus, the content stream becomes a full self-attention layer and sees all words in the original example. We do not apply the original input technique with context2vec and ELMo since, for these models, there is no reasonable representation that can be used to predict possible words at some timestep while depending on the input at that timestep, at least without fine-tuning. For other models, this option significantly outperforms target word masking and does not require many additional efforts. Hence, if not specified otherwise, we use it in our experiments with pure BERT, RoBERTa, XLNet estimators, and in +embs method when estimating $P(s|C)$ with BERT and XLNet.

3.2 Neural Language Models for Lexical Substitution

Different LMs/MLMs are employed as described below to obtain context-based substitute probability distribution $P(s|C)$. For each of them, we experiment with different target injection methods.

C2V Context2vec encodes left, and right context separately using its forward and backward LSTM layers correspondingly and then combines their outputs with two feed-forward layers producing the final full context representation. Possible substitutes are ranked by the dot product similarity of their embeddings and the context representation. We use the original implementation[3] and the weights[4] pre-trained on ukWac dataset.

ELMo To encode left and right context with ELMo, we use its forward and backward stacked LSTMs, which were pre-trained with LM objective. However, there are no pre-trained layers to combine their outputs. Thus, we obtain two independent distributions over possible substitutes: one given the left context $P(s|L)$, another given the right context $P(s|R)$. To combine these distributions we use BComb-LMs method proposed in (Arefyev et al., 2019), which can be derived similarly to +embs method describe above: $P(s|L,R) = \frac{P(s|L)P(s|R)}{P^{\gamma}(s)}$. The original version of ELMo described in (Peters et al., 2018) is used, which is the largest version pre-trained on 1B Word Corpus.

BERT/RoBERTa By default, we give our full example without any masking as input and calculate the distribution over the model's vocabulary at the position of the first subword of the target word. We employ BERT-large-cased and RoBERTa-large models as the best-performing ones. Unlike BERT and XLNet, we found that RoBERTa with +embs injection method performs better if cosine similarity instead of dot-product similarity is used when estimating $P(s|T)$ and the target word is masked when estimating $P(s|C)$. Thus, in the following experiments, we use these choices by default for RoBERTa+embs model.

XLNet By default, we use the XLNet-large-cased model with the original input, obtaining substitute probability distribution similarly to BERT. We found that for short contexts, XLNet performance degrades. To mitigate this problem, we prepend the initial context with some text ending with the end-of-document special token.

4 Baseline Lexical Substitution Models

Lexical substitution models described above are compared to the best previously published results, as well as our re-implementations of the following baseline models proposed in (Roller and Erk, 2016).

[3] https://github.com/orenmel/context2vec

[4] https://u.cs.biu.ac.il/~nlp/resources/downloads/context2vec/

OOC Out of Context model ranks possible substitutes by their cosine similarity to the target word and completely ignores given context. Following (Roller and Erk, 2016) we use dependency-based embeddings[5] released by (Melamud et al., 2015b).

nPIC Non-parameterized Probability In Context model returns the product of two distributions measuring the fitness of a substitute to the context and to the target: $nPIC(s|T,C) = P(s|T) \times P_n(s|C)$, where $P(s|T) \propto \exp(\langle embs_s, embs_T \rangle)$ and $P_n(s|C) \propto \exp(\sum_{c \in C} \langle embs_s, embs'_c \rangle)$. Here $embs$ and $embs'$ are dependency-based word and context embeddings, and C are those words that are directly connected to the target in the dependency tree.

5 Intrinsic Evaluation

We perform an intrinsic evaluation of the proposed models on two lexical substitution datasets.

5.1 Experimental Setup

Lexical substitution task is concerned with finding appropriate substitutes for a target word in a given context. This task was originally introduced in SemEval 2007 Task 10 (McCarthy and Navigli, 2007) to evaluate how distributional models handle polysemous words. In the lexical substitution task, annotators are provided with a target word and its context. Their task is to propose possible substitutes. Since there are several annotators, we have some weight for each possible substitute in each example, which is equal to the number of annotators provided this substitute.

We rank substitutes for a target word in a context by acquiring probability distribution over vocabulary on the target position. Lexical substitution task comes with two variations: candidate ranking and all-words ranking. In candidate ranking task, models are provided with a list of candidates. Following previous works, we acquire this list by merging all gold substitutes of the target lemma over the corpus. We measure performance on this task with Generalized Average Precision (GAP) that was introduced in (Thater et al., 2010). GAP is similar to Mean Average Precision, and the difference is in the weights of substitutes: the higher the weight of the word, the higher it should be ranked. Following (Melamud et al., 2015a), we discard all multi-word expressions from the gold substitutes and omit all instances that are left without gold substitutes.

In the all-vocab ranking task, models are not provided with candidate substitutes. Therefore, it is much harder task than the previous one. Models shall give higher probabilities to gold substitutes than to all other words in their vocabulary usually containing hundreds of thousands of words. Following (Roller and Erk, 2016), we calculate the precision of the top 1 and 3 predictions (P@1, P@3) as an evaluation metric for the all-ranking task. Additionally, we look at the recall of top 10 predictions (R@10).

The following lexical substitution datasets are used:

SemEval 2007 Task 10 (McCarthy and Navigli, 2007) consists of 2010 sentences for 201 polysemous words, 10 sentences for each. Annotators were asked to give up to 3 possible substitutes.

CoInCo or Concepts-In-Context dataset (Kremer et al., 2014) consists of about 2500 sentences that come from fiction and news. In these sentences, each content word is a separate example, resulting in about 15K examples. Annotators provided at least 6 substitutes for each example.

5.2 Results and Discussion

Comparison to previously published results Table 1 contains metrics for candidate and all-vocab ranking tasks. We compare our best model (XLNet+embs) with the best previously published results presented in (Roller and Erk, 2016), context2vec (c2v) model (Melamud et al., 2016) and BERT for lexical substitution presented in (Zhou et al., 2019). The proposed model outperforms solid models such as PIC, c2v, and substitute vector by a large margin on both ranking tasks. Nevertheless, (Zhou et al., 2019) reported better results than XLNet+embs in both lexical substitution tasks. In (Zhou et al., 2019), authors add a substitute validation metric that measures the fitness of a substitute to a context. It is

[5] http://www.cs.biu.ac.il/~nlp/resources/downloads/lexsub_embeddings

[6] We could not reproduce the results of (Zhou et al., 2019) and their code was not available.

Model	SemEval 2007			CoInCo		
	GAP	P@1	P@3	GAP	P@1	P@3
Transfer Learning (Hintz and Biemann, 2016)	51.9	-	-	-	-	-
PIC (Roller and Erk, 2016)	52.4	19.7	14.8	48.3	18.2	13.8
Supervised Learning (Szarvas et al., 2013b)	55.0	-	-	-	-	-
Substitute vector (Melamud et al., 2015a)	55.1	-	-	50.2	-	-
context2vec (Melamud et al., 2016)	56.0	-	-	47.9	-	-
BERT for lexical substitution (Zhou et al., 2019) [6]	**60.5**	**51.1**	-	**57.6**	**56.3**	-
XLNet+embs	**59.6**	**49.5**	**34.9**	**55.6**	**51.5**	**39.9**
XLNet+embs (w/o target exclusion)	59.6	0.4	26.0	53.5	2.5	30.0
XLNet+embs (w/o lemmatization)	59.2	38.3	27.5	53.2	34.5	27.1
XLNet+embs ($\mathcal{T} = 1.0$)	52.6	34.8	24.6	49.4	40.5	30.4

Table 1: Intrinsic evaluation of our best model and its variations on lexical substitution datasets.

computed as the weighted sum of cosine similarities of contextualized representations of words in two sentence versions: original and one where the target word is replaced with the substitute. This technique substantially improves predictions. However, substitute validation requires additional forward passes, hence, increasing computational overhead. Our methods need only one forward pass. Our approach is orthogonal to the substitute validation. Thus, a combination of two methods can improve results further. It is worth mentioning that BERT and XLNet work on a subword level. Hence, their vocabularies are much smaller in size (30K subwords) than those of ELMo (800K words) or C2V (180K words) and contain only a fraction of possible substitutes. Thus, these models can be significantly improved by generating multi-token substitutes.

Additionally, table 1 includes ablation analysis of our best model. Using ordinary softmax (which is equivalent to setting $\mathcal{T} = 1.0$) results in all metrics decreasing by a large margin. Also, post-processing of substitutes has a significant impact on all-words ranking metrics. Since LMs/MLMs generate grammatically plausible word forms and often generate the target word itself among top substitutes, additional lemmatization and target exclusion is required to match gold substitutes. In (Roller and Erk, 2016), the authors used the NLTK English stemmer to exclude all forms of the target word. In (Melamud et al., 2016) NLTK WordNet lemmatizer is used to lemmatize only candidates. For a fair comparison, the same post-processing is used for all models in the following experiments.

Model	SemEval 2007				CoInCo			
	GAP	P@1	P@3	R@10	GAP	P@1	P@3	R@10
OOC	44.65	16.82	12.83	18.36	46.31	19.58	15.03	12.99
nPIC	52.46	23.22	17.61	27.4	48.57	25.75	19.12	17.33
C2V	**55.81**	7.79	5.92	11.03	48.32	8.01	6.63	7.54
C2V+embs	53.40	**28.01**	**21.72**	**33.02**	**50.73**	**29.64**	**24.0**	**21.37**
ELMo	53.63	11.73	8.59	13.93	49.47	13.66	10.87	11.34
ELMo+embs	**54.15**	**31.95**	**22.20**	**31.82**	**52.22**	**35.92**	**26.6**	**23.80**
BERT	**54.40**	38.34	27.71	39.72	50.50	42.56	32.64	28.63
BERT+embs	53.88	**41.64**	**30.57**	**43.48**	**50.85**	**46.05**	**35.63**	**31.37**
RoBERTa	56.73	32.00	24.35	36.89	50.63	34.77	27.15	25.12
RoBERTa+embs	**58.83**	**44.13**	**31.67**	**44.70**	**54.68**	**46.54**	**36.33**	**32.03**
XLNet	59.10	31.70	22.80	34.90	53.39	38.16	28.58	26.46
XLNet+embs	**59.62**	**49.48**	**34.88**	**47.18**	**55.63**	**51.48**	**39.91**	**34.91**

Table 2: Comparison of our models and re-implemented baselines with the same post-processing.

Re-implementation of the baselines In Table 2, we compare our models based on different LMs/MLMs with and without +embs injection technique. Remember that BERT, RoBERTa, and XLNet see the target even if +embs is not applied, thus, providing already strong baseline results. All compared models, including re-implemented OOC and nPIC, employ the same post-processing consisting of substitute lemmatization followed by target exclusion. First, we notice that our best substitution models substantially outperform word2vec based PIC and OOC methods. For example, the XLNet+embs gives 2x better P@1 and P@3 than the baselines. This indicates that proposed models are better at capturing the meaning of a

word in a context as such, providing more accurate substitutes. On the candidate ranking task bare C2V model outperforms ELMo and BERT based models, but it shows the lowest Precision@1. We note that +embs technique substantially improves the performance of all models in all-vocab ranking task, and also increases GAP for the majority of models.

Injection of information about target word Next, we compare target injection methods described in Section 3. Figure 1 presents the Recall@10 metric for all of our neural substitution models with each applicable target injection method.

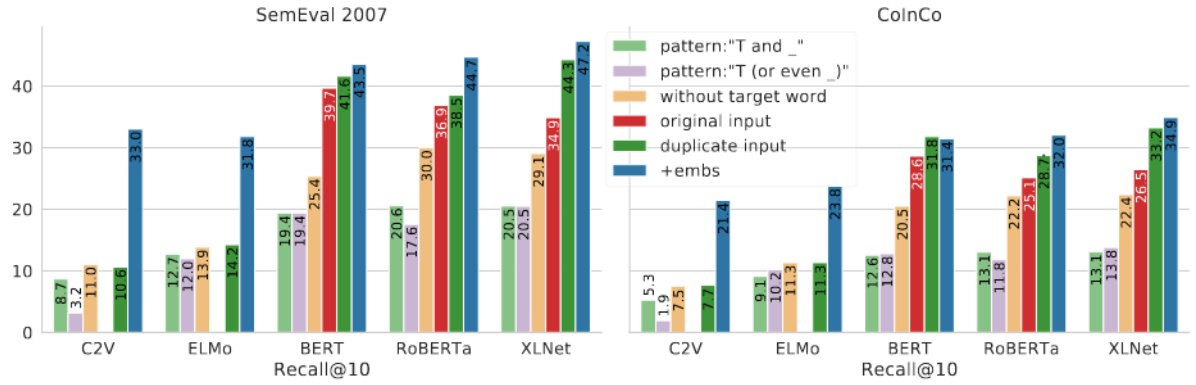

Figure 1: Comparison of various target information injection methods (SemEval 2007 and CoInCo datasets).

Application of dynamic patterns leads to lower performance even compared to the models that do not see the target word. Although we show the target word to the substitute generator, the pattern can spoil predictions, e.g., using "T and _" pattern with a verb can produce words denoting subsequent actions related to the verb, but not its synonyms. When we use the original input without any masking, the models produce substitutes more related to the target, resulting in good baseline performance. Applying the +embs method leads to a significant increase of Recall@10 for all models, almost always being the best performing injection method. For C2V and ELMo, it gives 2-3 times better recall than all other injection methods. Duplicating input performs surprisingly good for Transformer-based models but does not help for LSTM-based C2V and ELMo. This is likely related to the previous observations that Transformers are very good at copying words from the surrounding context, thus, predicting the original target word, but also words with similar embeddings. The highest impact is for the XLNet model as it can not straightforwardly use information from the target position due to its autoregressive nature but can easily find the target in the sentence copy. Overall, our experiments show that proper injection of the information about the target word results in much more plausible substitutes generated.

Hyperparameters Since there is no development set of reasonable size in SemEval 2007 dataset, we decided to select hyperparameters for SemEval 2007 on CoInCo and vice versa. However, the selected hyperparameters turned out to be the same. Thus, for both datasets we use $\mathcal{T}$=0.1 for BERT+embs, XLNet+embs and ELMo+embs models, $\mathcal{T}$=0.25 for RoBERTa+embs and $\mathcal{T}$=1.0 for C2V+embs. For BERT+embs, XLNet+embs, RoBerta+embs, C2V+embs β=0.0. For ELMo and ELMo+embs γ=0.5, β=1.5.

6 Extrinsic Evaluation

In this section, we perform the extrinsic evaluation of our models applied to the Word Sense Induction (WSI) task. The data for this task commonly consists of a list of ambiguous target words and a corpus of sentences containing these words. Models are required to cluster all occurrences of each target word according to their meaning. Thus, the senses of all target words are discovered in an unsupervised fashion. For example, suppose that we have the following sentences with the target word *bank*:

1. *He settled down on the river bank and contemplated the beauty of nature*,

2. *They unloaded the tackle from the boat to the bank.*

3. *Grand River bank now offers a profitable mortgage.*

Sentences 1 and 2 shall be put in one cluster, while sentence 3 must be assigned to another. This task was proposed in several SemEval competitions (Agirre and Soroa, 2007; Manandhar et al., 2010; Jurgens and Klapaftis, 2013). The current state-of-the-art approach (Amrami and Goldberg, 2019) rely on substitute vectors, i.e., each word usage is represented as a substitute vector based on the most probable substitutes, then clustering is performed over these substitute vectors.

Model	SemEval-2010 (AVG)	SemEval-2013 (AVG)
(Amrami and Goldberg, 2018)	–	25.43±0.48
(Amrami and Goldberg, 2019)	**53.6**±1.2	**37.0**±0.5
C2V	**38.9**	18.2
C2V+embs	28.5	**21.7**
ELMo	41.8	27.6
ELMo+embs	**45.3**	**28.2**
BERT	52.0	34.5
BERT+embs	**53.8**	**36.8**
RoBERTa	49.6	34.0
RoBERTa+embs	**51.4**	**34.5**
XLNet	52.2	33.4
XLNet+embs	**54.2**	**37.3**

Table 3: Extrinsic evaluation on word sense induction datasets.

We implemented a WSI algorithm using lexical substitutes from our models. The algorithm is a simplified version of methods described in (Amrami and Goldberg, 2019; Arefyev et al., 2019). In the first step, we generate substitutes for each example, lemmatize them and take 200 most probable ones. We treat these 200 substitutes as a document. Then, TF-IDF vectors for these documents are calculated and clustered using agglomerative clustering with average linkage and cosine distance. The number of clusters maximizing the silhouette score is selected for each word individually.

In table 3 we compare our lexical substitution models on two WSI datasets. For the previous SOTA models, which are stochastic algorithms, the mean and the standard deviation are reported. Our WSI algorithm is deterministic; hence, we report the results of a single run. Our best model achieves higher metrics than the previous SOTA on both datasets, however, the difference is within one standard deviation. Similarly to the intrinsic evaluation results, our +embs injection method substantially improves the performance of all models for WSI, except for the C2V+embs model on SemEval-2010, which probably used suboptimal hyperparameters.

Hyperparameters The optimal hyperparameters for WSI models were selected on the TWSI dataset (Biemann, 2012). For both evaluation datasets we used β=2.0 for BERT+embs, XLNet+embs, RoBERTa+embs and ELMo+embs models, β=0.0 for C2V+embs, $\mathcal{T}$=2.5 for BERT+embs, $\mathcal{T}$=1.0 for XLNet+embs, $\mathcal{T}$=10.0 for RoBERTa+embs, $\mathcal{T}$=0.385 for ELMo+embs and $\mathcal{T}$=1.0 for C2V+embs.

7 Analysis of Semantic Relation Types

In this section we analyze the types of substitutes produced by different neural substitution models.

7.1 Experimental Setup

For this analysis, the CoInCo lexical substitution dataset (Kremer et al., 2014) described above is used. We employ WordNet (Miller, 1995) to find the relationship between a target word and each generated substitute. First, from all possible WordNet synsets two synsets containing the target word and its substitute with the shortest path between them are selected. Then relation between these synsets is identified as follows. If there is a direct relation between the synsets, i.e. synonymy, hyponymy, hypernymy, or co-hyponymy with a common direct hypernym, we return this relation. Otherwise, we search for an indirect relation, i.e. transitive hyponymy or hypernymy, or co-hyponymy with a common hypernym at a distance of maximum 3 hops from each synset. We also introduce several auxiliary relations: *unknown-word* – the

target or the substitute is not found among WordNet synsets with the required PoS, *unknown-relation* – the target and the substitute are in the same WordNet tree, but no relation can be assigned among those described above, *no-path* – the target and the substitute are in different trees.

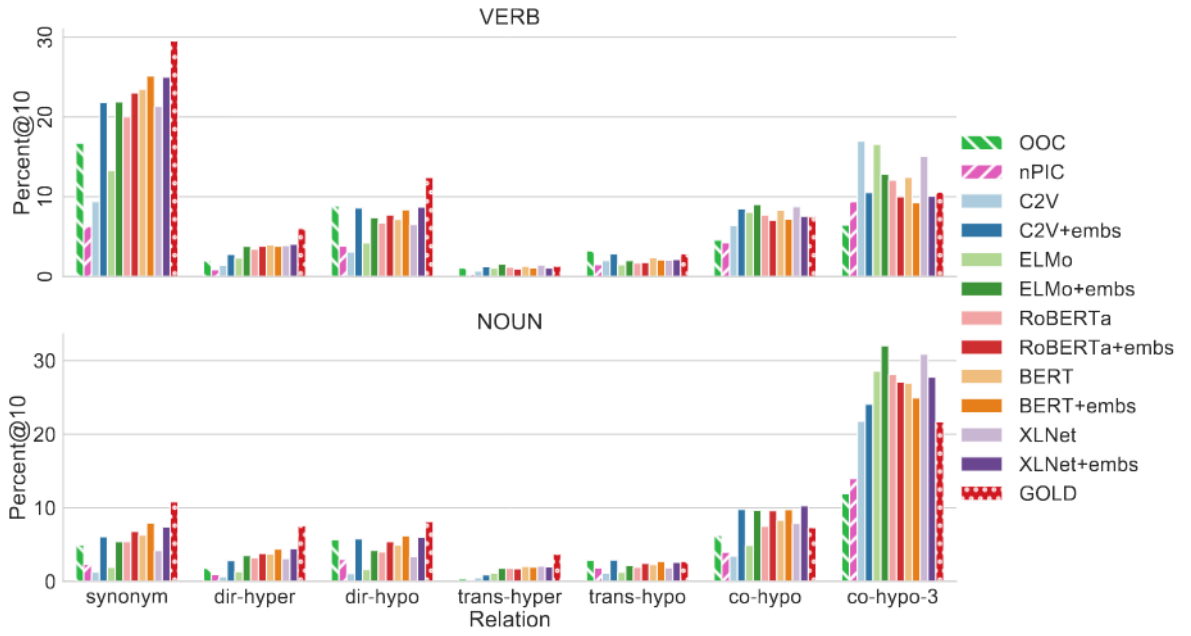

Figure 2: Proportions of substitutes related to the target by various semantic relations according to WordNet. We took top 10 substitutes from each model and all substitutes from the gold standard.

7.2 Discussion of Results

For nouns and verbs the proportions of non-auxiliary relations are shown in figure 2, for all words and relations see Appendix B. Our analysis shows that a substantial fraction of substitutes has no direct relation to the target word in terms of WordNet, even in case of the gold standard substitutes. Besides, even human annotators occasionally provide substitutes of incorrect PoS, e.g., for *bright* as an adjective there is the verb *glitter* among gold substitutes. For adjectives and adverbs 18% and 25% of gold substitutes are unknown words (absent among synsets with the correct PoS), while for verbs and nouns less than 7% are unknown. For baseline models OOC and nPIC, the overwhelming number of substitutes are unknown words. One of the reasons for this might be the fact that their vocabularies contain words with typos, but we also noticed that these models frequently do not preserve the PoS of the target word. The models based on LMs/MLMs produce much fewer unknown substitutes. Surprisingly, our +embs target injection method further reduces the number of such substitutes, achieving the proportion comparable to the gold standard. We can therefore suggest that our injection method helps to better preserve the correct PoS even for SOTA MLMs.

For both nouns and verbs, +embs target injection method consistently increases the proportions of synonyms, direct hyponyms and direct hypernyms, while decreasing the proportions of distantly related co-hyponyms (co-hypo-3) or unrelated substitutes. This is more similar to the proportions in human substitutes. Thus, the addition of information from embeddings forces the models to produce words that are more closely related to the target word and more similar to human answers. For C2V and ELMo, which have no information on the target word, target word injection results in 2x-3x more synonyms generated.

For several sentences from SemEval 2007 dataset (McCarthy and Navigli, 2007), Figure 3 shows some examples of substitutes provided by the human annotators (GOLD) and generated by several models, see Appendix A for more examples and models. The first example shows the case when +embs injection method improves the result, ranking closely related substitutes, such as *telephone*, *cellphone*, higher. The

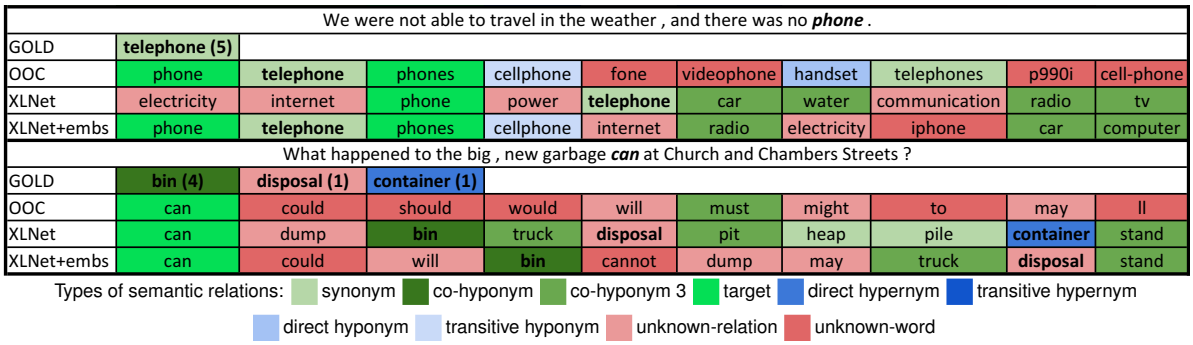

Figure 3: Examples of top substitutes provided by annotators (GOLD), the baseline (OOC), and two presented models (XLNet and XLNet+embs). The target word in each sentence is in bold, true positives are in bold also. The weights of gold substitutes are given in brackets. Each substitute is colored according to its relation to the target word. Substitutes before post-processing are shown.

substitutes provided by the bare XLNet model, such as *electricity* and *internet*, could be used in this context, but all the annotators had preferred the synonym *telephone* instead. The second case is related to the failure of the proposed method. Bare XLNet model generated substitutes related to the correct sense of the ambiguous target word *can*, and has all three gold substitutes among its top 10 predictions. In contrast, XLNet+embs produced words that are related to the most frequent sense: *will*, *could*, *cannot*, etc. We hypothesize that this problem could potentially be alleviated by choosing individual temperature for each example based on the characteristics of the combined distributions; this is a possible direction for our further research.

8 Conclusion

We presented the first comparison of a wide range of LMs/MLMs with different target word injection methods on the tasks of lexical substitution and word sense induction. Our results are the following: (i) large pre-trained language models yield better results than previous unsupervised and supervised methods of lexical substitution; (ii) if properly done, the integration of information about the target word substantially improves the quality of lexical substitution models. The proposed target injection method based on a fusion of a context-based distribution $P(s|C)$ with a target similarity distribution $P(s|T)$ proved to be the best one. When applied to the XLNet model, it yields new SOTA results on two WSI datasets. Finally, we study the semantics of the produced substitutes. This information can be valuable for practitioners selecting the most appropriate lexical substitution method for a particular NLP application.

Acknowledgements

We thank the anonymous reviewers for the valuable feedback. The contribution of Nikolay Arefyev to the paper was partially done within the framework of the HSE University Basic Research Program funded by the Russian Academic Excellence Project “5-100”.

References

Eneko Agirre and Aitor Soroa. 2007. SemEval-2007 task 02: Evaluating word sense induction and discrimination systems. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 7–12, Prague, Czech Republic, June. Association for Computational Linguistics.

Asaf Amrami and Yoav Goldberg. 2018. Word sense induction with neural biLM and symmetric patterns. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4860–4867, Brussels, Belgium, October-November. Association for Computational Linguistics.

Asaf Amrami and Yoav Goldberg. 2019. Towards better substitution-based word sense induction. *CoRR*, abs/1905.12598.

Nikolay Arefyev, Boris Sheludko, and Alexander Panchenko. 2019. Combining lexical substitutes in neural word sense induction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'19)*, RANLP '19, pages 62–70, Varna, Bulgaria.

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

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, June. Association for Computational Linguistics.

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

David Jurgens and Ioannis Klapaftis. 2013. SemEval-2013 task 13: Word sense induction for graded and non-graded senses. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 290–299, Atlanta, Georgia, USA, June. 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, Gothenburg, Sweden, April. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Suresh Manandhar, Ioannis Klapaftis, Dmitriy Dligach, and Sameer Pradhan. 2010. SemEval-2010 task 14: Word sense induction &disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 63–68, Uppsala, Sweden, July. Association for Computational Linguistics.

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

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

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

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, august.

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

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

Stephen Roller and Katrin Erk. 2016. PIC a different word: A simple model for lexical substitution in context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1121–1126, San Diego, California, june. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *AAAI*, pages 8766–8774.

Aina Garí Soler, Anne Cocos, Marianna Apidianaki, and Chris Callison-Burch. 2019. A comparison of context-sensitive models for lexical substitution. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 271–282, Gothenburg, Sweden, 23–27 May. Association for Computational Linguistics.

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

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

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

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*, pages 5753–5763.

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, July. Association for Computational Linguistics.

A Additional Examples of Lexical Substitutes

We were not able to travel in the weather , and there was no ***phone*** .										
GOLD	**telephone (5)**									
OOC	phone	**telephone**	phones	cellphone	fone	videophone	handset	telephones	p990i	cell-phone
C2V	let-up	anti-climax	rain	wind	excuse	sunshine	fog	snow	gridlock	respite
C2V+embs	phone	**telephone**	phones	cellphone	videophone	landline	voicemail	telephone/fax	telephones	email
ELMo	electricity	danger	snow	indication	sign	reason	rain	escape	word	doubt
ELMo+embs	phone	**telephone**	cellphone	phones	radio	electricity	e-mail	contact	computer	vehicle
BERT	phone	**telephone**	phones	radio	connection	car	line	cable	backup	mobile
BERT+embs	phone	**telephone**	phones	radio	cell	mobile	cable	car	call	connection
RoBERTa	phone	money	power	tv	**telephone**	cellphone	internet	food	job	car
RoBERTa+embs	phone	**telephone**	phones	cellphone	smartphone	internet	electricity	computer	radio	iphone
XLNet	electricity	internet	phone	power	**telephone**	car	water	communication	radio	tv
XLNet+embs	phone	**telephone**	phones	cellphone	internet	radio	electricity	iphone	car	computer
What happened to the big , new garbage ***can*** at Church and Chambers Streets ?										
GOLD	**bin (4)**	**disposal (1)**	**container (1)**							
OOC	can	could	should	would	will	must	might	to	may	'll
C2V	bins	guzzlers	collector	collection	dispenser	emporium	bucket	eaters	**bin**	cans
C2V+embs	can	could	cannot	will	would	should	might	must	*can*	can`t
ELMo	bins	trucks	bags	cans	truck	machines	dump	machine	collection	box
ELMo+embs	can	cannot	doesn	could	couldn	'll	must	don	should	might
BERT	can	**bin**	bag	lot	pile	heap	cans	bucket	closet	could
BERT+embs	can	could	**bin**	cans	bag	lot	will	may	don	bucket
RoBERTa	can	cans	**bin**	will	box	cart	**container**	could	lot	truck
RoBERTa+embs	can	could	cans	will	bins	may	**container**	**bin**	would	should
XLNet	can	dump	**bin**	truck	**disposal**	pit	heap	pile	**container**	stand
XLNet+embs	can	could	will	**bin**	cannot	dump	may	truck	**disposal**	stand
To fight each other is not a natural state for that dog or that ***bull*** to be in .										
GOLD	male bovine (1)	male cow (1)	bovine animal (1)	cow (1)						
OOC	bull	bulls	tup	dandie	**cow**	heifer	goat	nandi	tusker	comyn
C2V	dog	horse	ought	wants	person	needs	deserves	happens	thing	animal
C2V+embs	bull	horse	**cow**	fox	dog	stallion	bulls	cock	lion	pig
ELMo	person	situation	creature	horse	guy	dog	someone	beast	thing	else
ELMo+embs	bull	bear	horse	dog	animal	bulls	cat	pet	tiger	beast
BERT	bull	**cow**	bitch	beef	ass	devil	cock	ram	wolf	man
BERT+embs	bull	**cow**	ram	beef	tiger	cock	buffalo	bitch	horse	elephant
RoBERTa	bull	bulls	dog	buffalo	animal	man	**cow**	horse	human	other
RoBERTa+embs	bull	cat	dog	pig	horse	rabbit	puppy	wolf	bulls	donkey
XLNet	cat	horse	human	person	man	wolf	lion	pig	bear	**cow**
XLNet+embs	bull	horse	**cow**	lion	pig	cat	bear	tiger	wolf	goat
Anyway , I saw your ad on the net and just wanted to ***drop*** you a line to say hello .										
GOLD	write (3)	email (1)	send a message (1)	send (1)	text (1)	write a letter (1)	send a letter (1)			
OOC	drop	drops	dropping	dropped	drop-off	fall	decrease	decline	slump	pullback
C2V	give	**send**	drop	e.mail	ask	leave	post	tell	call	**email**
C2V+embs	drop	dropped	dropping	drops	pull	throw	give	**send**	jump	break
ELMo	tell	give	**send**	thank	ask	bring	**write**	get	throw	follow
ELMo+embs	drop	jump	**send**	pull	throw	bring	give	slip	slide	dip
BERT	drop	give	**send**	put	cut	dropped	tip	toss	leave	tell
BERT+embs	drop	drops	dropped	dropping	give	dip	toss	slip	cut	**send**
RoBERTa	drop	dropping	dropped	drops	pop	leave	throw	**write**	**send**	shoot
RoBERTa+embs	drop	dropped	dropping	drops	shoot	throw	give	slip	**send**	toss
XLNet	drop	**send**	give	shoot	**write**	throw	dropped	leave	blow	pop
XLNet+embs	drop	drops	dropped	dropping	throw	shoot	**send**	slip	give	fall

Types of semantic relations: synonym | co-hyponym | co-hyponym 3 | target | direct hypernym | transitive hypernym | direct hyponym | transitive hyponym | unknown-relation | unknown-word | no-path | multiword expression

Figure 4: Examples of substitutes produced by various lexical substitution methods based on original neural language models and their improved versions with +embs target word injection. Sentences are from SemEval 2007 dataset.

B Proportions of substitute types

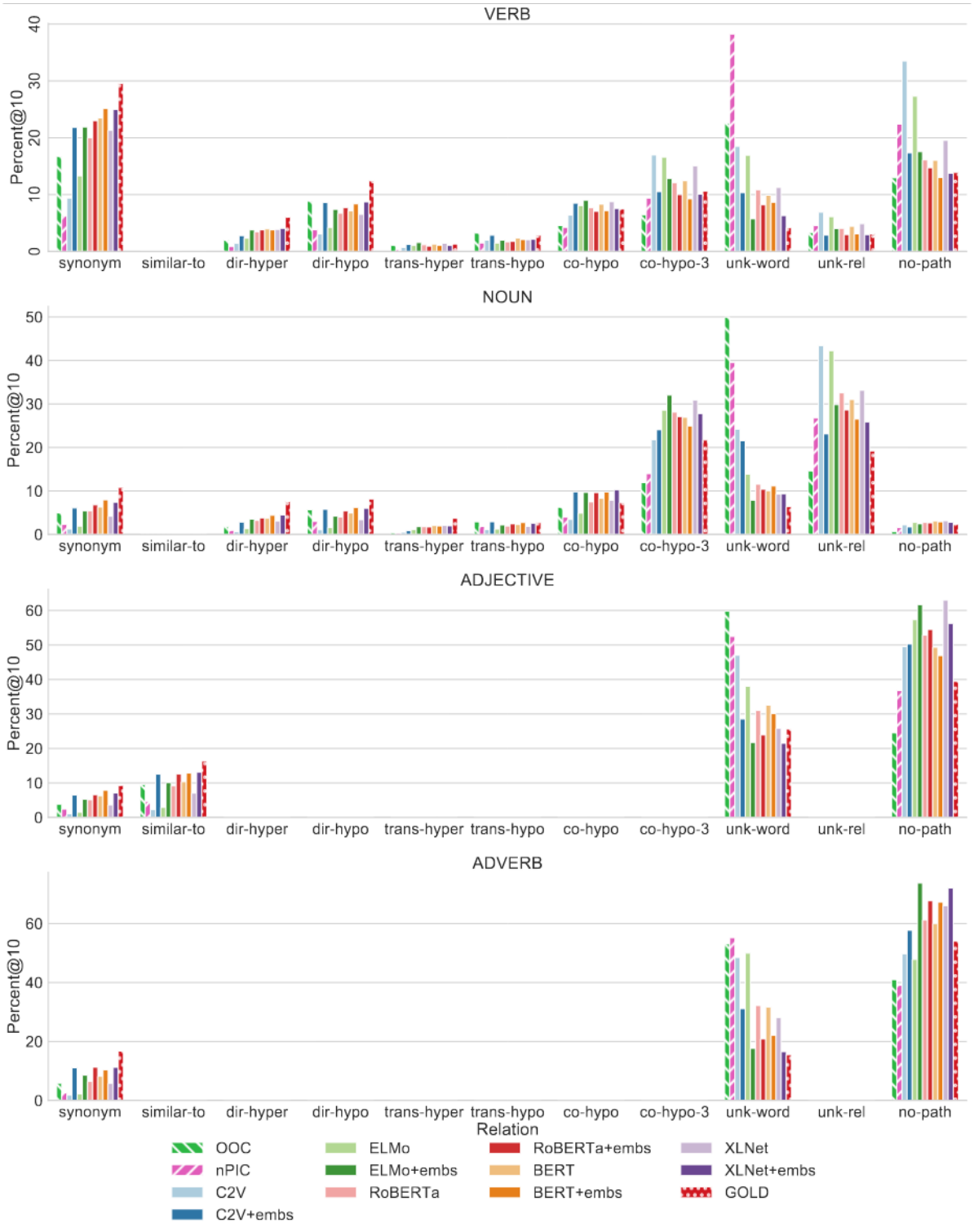

Figure 5: Proportions of substitutes related to the target by various semantic relations according to WordNet. We took top 10 substitutes from each model and all substitutes from the gold standard. Examples are from the CoInCo dataset.