ALaSca: an Automated Approach for Large-Scale Lexical Substitution

Caterina Lacerra[1] , **Tommaso Pasini**[2,*] , **Rocco Tripodi**[1] and **Roberto Navigli**[1]

[1] Department of Computer Science, Sapienza University of Rome
[2] Department of Computer Science, University of Copenhagen

{caterina.lacerra, rocco.tripodi, roberto.navigli}@uniroma1.it,
tommaso.pasini@di.ku.dkso

Abstract

The lexical substitution task aims at finding suitable replacements for words in context. It has proved to be useful in several areas, such as word sense induction and text simplification, as well as in more practical applications such as writing-assistant tools. However, the paucity of annotated data has forced researchers to apply mainly unsupervised approaches, limiting the applicability of large pre-trained models and thus hampering the potential benefits of supervised approaches to the task. In this paper, we mitigate this issue by proposing ALaSca, a novel approach to automatically creating large-scale datasets for English lexical substitution. ALaSca allows examples to be produced for potentially any word in a language vocabulary and to cover most of the meanings it lists. Thanks to this, we can unleash the full potential of neural architectures and finetune them on the lexical substitution task. Indeed, when using our data, a transformer-based model performs substantially better than when using manually-annotated data only. We release ALaSca at https://sapienzanlp.github.io/alasca/.

1 Introduction

The lexical substitution task [McCarthy and Navigli, 2009] requires a system to provide possible replacements for a target word in a given sentence. Each proposed substitute should fit the context, while maintaining the overall meaning of the sentence unchanged. Through the years, two sub-tasks have been put forward, i.e., *substitutes prediction* and *candidates ranking* [Melamud *et al.*, 2015]. The former aims at producing one or more possible substitutes given the target word and its context; the latter, instead, ranks a set of substitutes provided in advance to the system. Neither of these task variants explicitly requires any semantic tagging of targets or substitutes; however, a good substitution system is expected to implicitly take word senses into account. Consider, for example, two contexts for the target *bright*: a) She is a *bright* girl; b) The Sun is *bright*. An effective substitution system would provide two distinct sets of possible replacements, such as {*smart*, *intelligent*, *brilliant*} and {*shining*, *luminous*}, respectively. The implicit disambiguation provided by substitution systems has shown itself to be useful in several fields, such as word sense induction [Başkaya *et al.*, 2013; Amrami and Goldberg, 2018; Arefyev *et al.*, 2019], text augmentation [Jia *et al.*, 2019; Arefyev *et al.*, 2020], word sense disambiguation [Hou *et al.*, 2020] or text simplification [Bingel *et al.*, 2018]. However, despite its possible uses, there is a lack of appropriate large-scale resources for the task [Soler *et al.*, 2019]. This prevents exploitation of the most recent advances in neural language modeling such as Bidirectional Encoder Representations from Transformers [Devlin *et al.*, 2019, BERT] or Generative Pre-trained Transformer [Radford *et al.*, 2019, GPT], which hampers the potential benefits they could bring to this field.

To fill this gap, we propose ALaSca, an Automated approach for Large-Scale lexical substitution. ALaSca produces datasets tailored for the task, automatically associating target words in context with ranked lists of substitutes. Starting from an arbitrary target word, it first extracts Wikipedia sentences where this word appears. Then, by leveraging latent representations of texts, ALaSca retrieves new contexts that are similar to those where the target appears and uses them to find meaningful substitutes. Finally, these substitutes are ranked according to how well they match the initial input context. ALaSca makes it possible, for the first time, to create a large number of examples annotated with lexical substitutes that are consistent with the context. By exploiting a cluster-based approach, we enforce a diversity of contexts and therefore of meanings in which the input words are used across sentences. The resulting dataset enables neural architectures to be finetuned on the task, unleashing the full potential of supervised techniques. Through different experiments, we show that a simple BERT-based model provides better substitutes when using ALaSca training data than when being restricted to gold instances only, reaching performances that are higher, or on par with, complex state-of-the-art models. We also investigate the reasons for this success and provide an extensive ablation study over the several components of ALaSca, measuring to what extent each of them influences the quality of the resulting dataset.

[*] Work carried out while at the Sapienza University of Rome.

2 Related Work

The Lexical Substitution task aims at finding the most suitable replacements for a target word in a given context without substantially changing the overall meaning of the sentence [McCarthy and Navigli, 2009]. In stark contrast with many other NLP problems, most approaches to this task have to date been knowledge-based or unsupervised [Melamud *et al.*, 2015; Melamud *et al.*, 2016; Zhou *et al.*, 2019], due to the lack of large-scale resources needed to finetune pretrained models for text understanding.

2.1 Lexical Substitution Datasets

Over the years, various resources have been released for the English lexical substitution task.

LST The Lexical Substitution Task dataset (LST) is the dataset released for the original task proposed by McCarthy and Navigli [2007]. It encompasses 2010 sentences, with a single word annotated for each sentence. The sentences were drawn from the English Internet Corpus [Sharoff, 2006], and cover 201 distinct target words, balanced across different parts of speech. While targets and sentences were partially collected using an automatic approach, the substitutes were chosen manually by 5 English native speakers.

TWSI The small coverage of the LST dataset led to the creation of the Turk bootstrap Word Sense Inventory [Biemann, 2012], which was the first attempt to produce a large-scale dataset. Biemann collected roughly 25K Wikipedia sentences, and tagged 1012 distinct nouns therein through Amazon Mechanical Turk. Furthermore, the resource annotation was carried out by considering the meaning of each target word occurrence, thereby providing different substitutes for the same lemma in different contexts.

CoInCo Despite the effort expended to create the TWSI, the corpus covered only nouns, hence, Kremer *et al.* [2014] proposed a similar annotation task starting from a set of sentences of the MASC corpus [Ide *et al.*, 2008]. Substitutes were annotated through Amazon Mechanical Turk in this case too and the resulting dataset (Concept In Context, CoInCo) contains 15K tagged instances in 2474 sentences for 3874 distinct words with diverse part-of-speech tags.

2.2 Lexical Substitution Models

The creation of several datasets for the task allowed the development of different strategies to tackle it. Unfortunately, the use of supervised models was limited to a few examples [Szarvas *et al.*, 2013a; Szarvas *et al.*, 2013b], due to the paucity of annotated data and the difficulty of creating them. The models proposed during the years can be divided in three categories: knowledge-based, vector-space, and Transformer-based.

Knowledge-based models These approaches exploit the structure of lexical resources, such as WordNet [Miller, 1995], to find the synonyms of a target word to be used as substitutes. Szarvas *et al.* [2013a] proposed a hybrid approach where a binary classifier is trained with delexicalized features to predict whether a retrieved substitute is valid in a given context. The main limitation of these models is that they may overlook good substitutes if they are not synonyms or if they are absent from the lexical resource used.

Vector-space models These approaches drop the need for a knowledge base and, exploiting word-embedding models, compute similarity scores among target words, contexts and substitutes. Melamud *et al.* [2015] proposed several similarity measures to select substitutes and later further extended their work by using a bi-LSTM to obtain word representations [Melamud *et al.*, 2016; Soler *et al.*, 2019]. This line of research developed representations of the context where a lexical item appears and were fundamental to the development of modern contextualized word embedding models [Devlin *et al.*, 2019].

Transformer-based models These models represent the natural evolution of vector-space models. They employ large pre-trained neural architectures to produce contextualized representations of words. Zhou *et al.* [2019] proposed a BERT-based approach consisting of two steps: first, it applies dropout to the target word and proposes candidates substitutes by means of a language modeling head. Second, these candidates are validated according to their substitution's influence on the contextualized representation of the sentence. Arefyev *et al.* [2020], instead, focused on developing substitute probability estimators that were then applied to several pretrained contextualized models.

In our work we address the training data paucity issue: rather than manually creating a dataset (see §2.1) we focus, for the first time, on the automatic creation of high-quality training data for the lexical substitution task. Differently from all the other approaches which produce annotated datasets by manually associating a list of substitutes for a given word in context, ALaSca starts from a list of lemmas and, first, builds a large set of examples where those lemmas appear in as many senses as possible, and then annotates them with their most suitable substitutes. Our method addresses most shortcomings of the existing resources as it is completely automatic. Hence it is capable of building large datasets with no human effort, while covering most of the senses of the input target words and potentially all words in a language vocabulary. At the same time, the automatic collection of substitutes does not invalidate the quality of our dataset, allowing a simple baseline model to achieve better results than when trained on manually-curated resources.

3 ALaSca

In this Section, we detail our approach to producing data for the lexical substitution task. Specifically, starting from an arbitrary set of target words, ALaSca performs three steps: i) it collects sentences where the input target words appear, clusters and samples them to ensure context heterogeneity (§3.1), ii) it extracts the candidate substitutes for target words in each sentence (§3.2), iii) it associates sentences and substitutes so as to build the final dataset (§3.3). A complete example of the ALaSca pipeline is depicted in Figure 1.

3.1 Retrieving, Clustering and Sampling

We take as input a set L of lexemes represented as pairs (l, pos), where l is a lemma and pos its part of speech. Then,

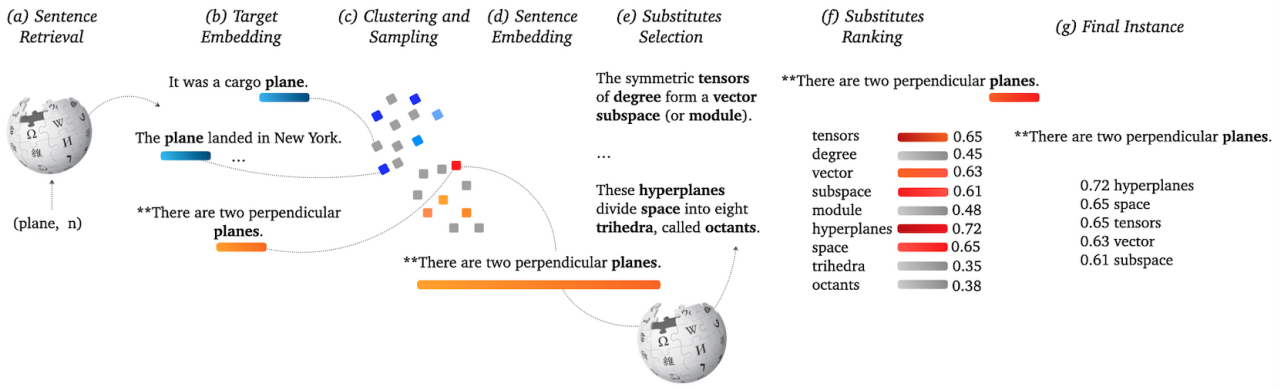

Figure 1: The ALaSca pipeline for the target noun *plane*. ALaSca retrieves target occurrences from a corpus (a), embeds them (b), clusters and samples them (c). Then, for each sampled sentence (**), it computes a sentence embedding (d) to be used to retrieve similar contexts (e) and ranks all the content words therein (f). Finally, we filter the substitutes obtained and associate them with the sampled sentence (g).

we retrieve a set of sentences S from a corpus C where at least one of the target lexemes in L appears in any of its inflected forms[1] (Figure 1 (a-b)). We define $S_l \subset S$ as the set of sentences where $l \in L$ appears. Then, we extract from S_l a subset of sentences that can be representative of all the meanings of l. Note that randomly sampling sentences from S_l would yield a set of examples where, most of the time, l appears with its most frequent meaning. This would limit our approach to producing substitutes for a few senses only. Thus, to ensure sense diversity, we first leverage a pretrained language model in order to produce a contextualized vector representation for each occurrence of l in S_l, and then cluster the resulting vectors. Hence, we can group instances of the same word in similar contexts, which therefore express similar meanings within the same cluster. Formally, for each sentence $s \in S_l$ represented as a sequence of tokens $s_1, \ldots, s_n$, let i be the position of the lemma l in the sentence s. We compute the vector representation of l in s as follows:

$$\mathbf{v}_{s,l} = NLM(s)[i] \quad (1)$$

where NLM is a generic Neural Language Model, and $NLM(s)[i]$ is the vector representation of the i-th token in s (Figure 1 (b)). We then cluster the resulting vectors by employing the k-means algorithm, and propose two strategies for determining k.

Fixed *k* In this variant, we fix the number of clusters k_l for each target $l \in L$. Since different meanings of the same lemma may actually be very similar to each other [Erk *et al.*, 2009], we set k_l to twice the senses enumerated in our reference sense inventory for lemma l, so as to have enough clusters to also capture subtle differences. To enumerate the possible senses of a word we rely on WordNet, a fine-grained lexical resource that defines senses as sets of synonym words, and is widely used as English sense inventory for lexical semantics tasks.

Adaptive *k* Fixing the number of clusters based on a manual resource may result in sub-optimal groupings. Thus, we also propose determining the number of clusters automatically by performing successive iterations of the k-means algorithm while varying the value for k_l. We choose the k_l leading to the best clustering in terms of silhouette score [Rousseeuw, 1987], i.e., a measure of cluster quality that considers at the same time how similar elements within the same cluster are and how distant each of them is from other clusters' elements.

In both variants, we retain the t elements of each cluster that are the closest to its center and sample d items among them (Figure 1 (c)). Finally, we map the sampled vectors to their original sentences and create the set S_l^{sample} of all subsampled sentences for l.

3.2 Substitutes Selection

In this step, for each sentence in S_l^{sample}, we retrieve its most similar sentences from a corpus C where l does not appear, and then extract the possible substitutes for l. Formally, let $F_{-l} = \{f \mid f \in C, l \notin f\}$ be the set of sentences where l does not appear. We compute a sentence embedding $\phi_s = SENT(s)$ and $\phi_f = SENT(f)$ for each sentence $s \in S_l^{sample}$ and $f \in F_{-l}$, respectively, by means of a generic sentence embedding model $SENT$ (Figure 1 (d)). For each sentence s in S_l^{sample}, we use its vector representation as a query to find the most similar sentences in F_{-l}. That is, we compute the set $M_s = \{f_1, \ldots, f_m \mid f_i \in F_{-l}\}$ of the m sentences most similar to s according to their vectors' cosine similarity (Figure 1 (e)). Finally, for each sentence f in M_s, we extract the set of candidate substitutes CS_s^l by considering all the words w of the sentences in M_s with the same part of speech of l (Figure 1 (e), bold tokens). More formally, $CS_s^l = \{(w, f) \mid \exists\, f \in M_s, w \in f, pos_l = pos_w\}$. Note that not all the words with the same pos as l in the sentences of M_s represent a suitable substitute for l. Hence, we devise two strategies for filtering them out and compile the final ranking of substitutes.

Contextualized similarity For each pair (w, f) in CS_s^l, we compute the contextualized embedding vectors $\mathbf{v}_{f,w}$ (Eq. 1), and their cosine similarity with the vector $\mathbf{v}_{s,l}$ (Figure 1(f)). Since CS_s^l may contain several inflected forms of the same

[1]For ease of reading, in what follows we consider a lemma l to occur in a sentence s if any of its inflected forms appears therein.

Clustering	S-Embeddings	Sim. Measure	γ	GAP
fixed	LASER	context	0.6	35.83
adaptive	LASER	context	0.6	33.54
fixed	S-BERT	context	0.6	34.92
adaptive	S-BERT	context	0.6	34.61
fixed	LASER	combined	0.4	**36.19**
adaptive	LASER	combined	0.4	32.85
fixed	S-BERT	combined	0.4	35.30
adaptive	S-BERT	combined	0.4	34.76

Table 1: The results on the development set for the ablation study.

lemma, we lemmatize each word therein, and assign to each lemma a score that is the average of its inflected forms scores. This yields a list of unique lemmas that we rank in descending order of similarity and from which we select up to α lemmas with a similarity score higher than a given threshold γ, thus obtaining the list of substitutes $R_{s,l}$ for the target lemma l in s (Figure 1(g)).

Combined similarity Since all similarity measures applied so far already take contexts into account, we propose enriching the contextualized similarity with a static metric, i.e., the cosine similarity between the source and candidate static word embeddings. Formally, let w be a candidate in CS_s^l, $\mathbf{c_w} = EMB(w)$ its static embedding and $\mathbf{c_l} = EMB(l)$ the static embedding of l, we compute the score for each candidate pair $(w, f) \in CS_s^l$ as follows: $score(l, w) = \frac{1}{2}(cos\_sim(\mathbf{v}_{s,l}, \mathbf{v}_{f,w}) + cos\_sim(\mathbf{c_l}, \mathbf{c_w}))$, and compute the final list of candidate lemmas $R_{s,l}$ as previously described.

3.3 Dataset Construction

So far, we have retrieved, for each lexeme l, a set of sentences S_l^{sample} where l occurs, and, for each sentence therein, we have computed a ranked list $R_{s,l}$ of substitutes. Thus, we can now create a dataset for each input lemma l by associating each sentence $s \in S_l^{sample}$ with its set of substitutes $R_{s,l}$. Note that, according to the different contexts where a target word appears, we have different sets of substitutes.

4 Model and Parameter Selection

In order to evaluate how each component of the pipeline influences the quality of the dataset, we perform an ablation study, analyzing the contribution of the clustering type (§3.1), the sentence encoder, the similarity measure and the similarity thresholds α and γ (§3.2). For each combination of these parameters, we first produce a dataset using ALaSca to train the reference model and then analyze the model's performance on a development set. We evaluate our model on the *candidates ranking* subtask, which requires a set of given target substitutes to be ranked according to the input context.

4.1 Setup

Reference model We use a simple yet effective BERT encoder with a language modeling head on top, as reference model, namely BERT$_{ft}$. It takes as input a sentence s where a target word w appears, encodes each BPE and feeds the average vectors of the target words' BPEs to the language modeling head. This ensures that we provide a single distribution d_w over the vocabulary for the given input word. However, since the task requires only the set of candidates provided for the target lemma to be ranked, we project d_w over these candidates and apply the softmax so as to have a distribution. The set of possible candidates, instead, is built by considering all the possible substitutes for the target lemma within the data utilized for training.

Development set As development, we use 30% of TWSI instances concatenated with the development split of CoInCo and we exclude the target words occurring in the development set from the data considered for training.

4.2 Results

In Table 1 we report the performance obtained when varying the sentence embedding model and the similarity measure, while keeping the similarity thresholds α and γ fixed[2].

Clustering We compare the *adaptive* and *fixed* clusterings that we detailed in §3.1. As shown in Table 1, the clustering with a *fixed* number of clusters always leads to higher results than the *adaptive* one. This is somehow to be expected, since the *fixed* approach provides on average a higher number of clusters than the *adaptive* one (8.30 and 2.66 average clusters per word, respectively) for the same set of vectors, leading to smaller and intuitively more coherent groupings of word occurrences. In contrast, having larger clusters leads to a sampling of the sentences that could collect less representative instances of the diverse meanings of the target word, thereby introducing more noise into the final dataset.

Sentence embeddings To extract information from the various sentences that may be useful for mining substitutes (§3.2), we experiment with two encoders: LASER [Artetxe and Schwenk, 2019] and Sentence BERT [Reimers and Gurevych, 2019, S-BERT]. This parameter does not strongly affect the performance; indeed, LASER leads the model to higher performance than S-BERT when the *fixed* clustering is used, while S-BERT helps to build a better dataset than LASER when using an *adaptive* number of clusters.

Similarity measure To rank the candidate substitutes (§3.2), we consider the contextualized similarity alone or in combination with the cosine similarity between static word embeddings. In all the configurations, the combined similarity leads to the creation of better-performing datasets, with the sole exception of one configuration: this is when LASER embeddings are used with an *adaptive* setting.

Similarity thresholds We tuned the thresholds α and γ for the selection of the substitute candidates (§3.2). The cosine similarity threshold, γ, ranges across $\{0.45, 0.50, 0.55, 0.60, 0.65\}$ when it is used with contextualized vectors and across $\{0.35, 0.40, 0.45, 0.50, 0.55\}$ when it is used with the combined similarity. We employ two different sets of values to take into account the naturally lower scores resulting from the average of the two similarities. We first tune γ, and then vary α across $\{1, 3, 5, 10\}$, so as to adjust the number of substitutes to retain. In Table

[2]A list of all the parameter combinations together with the corresponding results is available at https://sapienzanlp.github.io/alasca/.

Model	Dataset	GAP score
BERT [Arefyev *et al.*, 2020]	-	54.4
XLNet + embs [Arefyev *et al.*, 2020]	-	59.6
BERT for lexical substitution [Zhou *et al.*, 2019]	-	**60.5**
BERT_{unsup}	-	53.7
BERT_{ft}	CoInCo_T	56.2
BERT_{ft}	TWSI_T	58.7
BERT_{ft}	CoInCo_T + TWSI_T	59.6
BERT_{ft}	ALaSca_T	58.2
BERT_{ft}	ALaSca_T + CoInCo_T	59.3
BERT_{ft}	ALaSca_T + TWSI_T	59.8
BERT_{ft}	ALaSca_T + CoInCo_T + TWSI_T	**60.5**

Table 2: Results for the *candidates ranking* task on the LST test set.

Dataset	Target Words	Sentences	Instances	AVG Substitutes per Target
LST	201	2010	201	20.44
CoInCo	3874	2457	15629	22.75
TWSI	1012	24612	24644	50.24
CoInCo_T	3120	2434	14329	22.75
TWSI_T	947	22818	22842	58.20
ALaSca_T	3442	34755	37467	11.71

Table 3: Quantitative descriptions of our dataset, compared to the gold-standard corpora and the corresponding training splits.

1 we report the results setting $\alpha = 1$ and consider $\gamma = 0.6$ and $\gamma = 0.4$, respectively, for the contextualized and the combined similarity measures.

ALaSca setting For the remaining experiments, we use BERT large-cased as Neural Language Model, using the sum of its last four layers to embed the target words during the sentence sampling step (§3.1), and the last hidden state only for the substitutes ranking step (§3.2). In both cases, we encode a word by averaging the embedding of its sub-tokens. In light of the results attained in the ablation study, we use the *fixed* clustering, the LASER model to encode sentences and the combined similarity as proximity measure. As static word representation we leverage ConceptNet Numberbatch vectors [Speer and Lowry-Duda, 2017], and set $\gamma = 0.4$ and $\alpha = 1$.

5 Lexical Substitution Experimental Setup

In this Section, we set up of the experiments in the *candidates ranking* task to assess ALaSca dataset quality.

Gold training sets We consider the training split of CoInCo (CoInCo_T) and TWSI_T, i.e., the 70% of TWSI instances that we did not use for development.

ALaSca dataset To generate a dataset for training, we feed ALaSca with the list of lemmas in CoInCo [Kremer *et al.*, 2014] and LST [McCarthy and Navigli, 2009], using Wikipedia (December 2019 dump) as corpus to retrieve the sentences. We set the number of sentences $|S_l|$ to be retrieved from Wikipedia to 1000, from which we sample $d = 1$ examples from among the $t = 500$ closest to the centroids. Then, for substitutes selection, we set $m = 1000$ (§3.2). This setting is chosen so as to result in a dataset with a number of instances comparable to those of the CoInCo_T and TWSI_T concatenation, i.e. $40,000$, while at the same time assuring at least one sentence for each of the 3442 lexemes covered.[3] We refer to the resulting dataset as ALaSca_T and report its statistics in comparison to those of other gold datasets in Table 3. As one can see, ALaSca displays a wider variety of sentences for a number of instances comparable to that in the two gold corpora. ALaSca has been generated with this size to set a level playing field with other datasets, but it can be extended to cover the lemmas or number of instances desired.

Training parameters We train the reference model (§4.1) with the Kullback–Leibler divergence loss with RAdam [Liu *et al.*, 2019] and learning rate 10^{-5}. We set the maximum epochs to 5, with early stopping and patience set to 3.

Comparison systems We consider as comparison the same reference model trained on CoInCo_T and TWSI_T, as well as its unsupervised version, i.e., BERT_{unsup}. We also compare with the work of Arefyev *et al.* [2020], which proposes several unsupervised models for the task. Specifically, we consider their best-scoring model employing XLNet [Yang *et al.*, 2019] and the model that is the most similar to ours, i.e., the one based on BERT. Finally, we compare with the unsupervised BERT-based model reported by Zhou *et al.* [2019]. This latter model employs both the hidden state of the target word and the attention scores associated with it in order to rank the candidates for the target.

Evaluation We use the Lexical Substitution Task (LST) Dataset [McCarthy and Navigli, 2007] to carry out the evaluation. As standard for *candidates ranking*, we employ the Generalized Average Precision [Kishida, 2005, GAP] to compare the rankings produced by a model with the gold ones. The higher a substitute is ranked in the gold standard, the more this measure rewards its correct positioning in the rank-

[3] Due to the use of similarity thresholds, some input targets may be associated with empty sets of substitutes, thus being discarded.

Model	Dataset	GAP score
BERT$_{ft}$	TWSI$_T$ + CoInCo$_T$	58.83
BERT$_{ft}$	ALaSca$_T$	59.00
BERT$_{ft}$	ALaSca$_T$ + TWSI$_T$ + CoInCo$_T$	59.96

Table 4: Results for the zero-shot setting on the LST test set.

ing that is produced. The GAP score is formally defined as:

$$GAP = \frac{\sum_{i=1}^{N} I(x_i)p_i}{\sum_{i=1}^{R} I(y_i)\overline{y_i}} \qquad p_i = \frac{\sum_{k=1}^{i} x_k}{i}$$

where x_i is the gold standard weight of the i-th item as ranked by the model, $I(x_i)$ is a binary function that returns 1 if x_i is in the gold, else 0, R and N the sizes of the gold and the rankings produced, respectively, and $\overline{y_i}$ is the average of the gold weights for the i-th ranking. As standard in the literature [Arefyev *et al.*, 2020; Melamud *et al.*, 2015] we exclude multi-words substitutes from the gold and discard instances that do not have any substitutes left.

6 Results

In what follows, we present the results attained when training the reference model (§4.1) on different combinations of the datasets, i.e., CoInCo$_T$, TWSI$_T$ and ALaSca$_T$, and tested on the candidates ranking and the zero-shot settings.

Candidates ranking As first test, we use the standard setting for the candidates ranking task and report the results on the LST test set in Table 2. First, we note that finetuning on the task, regardless of the dataset, is always beneficial. Indeed, the finetuned versions of our reference model (middle block) always outperform the unsupervised version by from 2 to almost 6 points. While this was somehow to be expected, we are the first to show the effectiveness of finetuning in the lexical substitution task. When using ALaSca$_T$ data, which we recall are automatically-generated, we attain results that are either higher than, or in the same ballpark as, those attained when using each manually-annotated training set (CoInCo$_T$ and TWSI$_T$). This is an important result *per se* as it already shows that the data generated by our approach are comparable to those annotated manually. Concatenating ALaSca$_T$ with gold standards always results in performance improvements, and enables the model to reach state-of-the-art results when trained on the concatenation of the three datasets. Our model also surpasses its closest competitors, i.e., BERT$_{unsup}$ and the BERT model proposed by Arefyev *et al.* [2020], by 5 and 6 points, respectively. Furthermore, our reference model surpasses the best of the approaches proposed by Arefyev *et al.* [2020], which combines the probability of a substitute given the context, as obtained from the XLNet model, with the proximity of the target to the substitute, given by a tuned temperature softmax on the inner product of the substitute and the XLNet target embeddings. Finally, we compare to the model of Zhou *et al.* [2019], even though both Arefyev *et al.* [2020] and ourselves fail to reproduce their results. We attain their same result using a simpler model, thereby demonstrating that producing data automatically can effectively help in reducing model complexity.

Zero-shot Since ALaSca$_T$ also adds examples for target words of the test set, it is natural to ask whether the improvements are the result of testing only on words that were seen at training time. To investigate this possibility, we build a reduced training set by removing all the test targets from the training corpus, and finetune the model on the remaining instances only. As one can see from Table 4, ALaSca$_T$ data allows the model to generalise well on unseen words, better than when using the gold training data alone. Indeed, BERT's performance remains close to that attained when test targets are included in the training (see Table 2) and 1 point higher than when using TWSI$_T$ and CoInCo$_T$ data only.

Discussion The proposed dataset boosts the performance of a simple BERT-based model, and leads it to attain state-of-the-art performance. The chosen architecture has the advantage of being extremely simple, without any tricky computations [Zhou *et al.*, 2019] or dedicated injection of the target [Arefyev *et al.*, 2020](XLNet). Furthermore, ALaSca enables the amount of training data with annotations to be enlarged for arbitrary words. As already noted by Arefyev *et al.* [2020], BPE tokenization inherently limits the set of possible substitutes to only those that are not split into multiple subwords. On average, it is not possible to predict 40% of candidate substitutes for 50% of LST test instances, leaving space for future improvements of models for the task.

7 Conclusion

In this work, we proposed ALaSca, an Automated approach for Large-Scale lexical substitution. ALaSca is the first approach to automatically create silver data for the lexical substitution task, mitigating the lack of annotated data in this area. By leveraging a clustering approach, it takes different meanings of a word into account, and provides distinct sets of possible substitutes depending on the context the target word appears in. Thanks to the large datasets that can now be created, we can finetune pretrained language models, which thus far could be employed in an unsupervised fashion only. By finetuning a simple baseline model, we attain performances that are 6 points higher than comparable unsupervised approaches. Furthermore, our approach can create new datasets on demand, therefore allowing a model to be specialized on a specific domain or existing training data to be enriched with annotations for rare words. As future work, we will investigate possible applications of ALaSca to other languages and in downstream applications, e.g., text simplification.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487 under the European Union's Horizon 2020 research and innovation programme.

This work was supported in part by the MIUR under grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of the Sapienza University of Rome.

References

[Amrami and Goldberg, 2018] Asaf Amrami and Yoav Goldberg. Word sense induction with neural biLM and symmetric patterns. In *Proc. of EMNLP*, 2018.

[Arefyev *et al.*, 2019] Nikolay Arefyev, Boris Sheludko, and Alexander Panchenko. Combining lexical substitutes in neural word sense induction. In *Proc. of RANLP*, 2019.

[Arefyev *et al.*, 2020] Nikolay Arefyev, Boris Sheludko, Alexander Podolskiy, and Alexander Panchenko. Always keep your target in mind: Studying semantics and improving performance of neural lexical substitution. In *Proc. of COLING*, 2020.

[Artetxe and Schwenk, 2019] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *TACL*, 7, 2019.

[Başkaya *et al.*, 2013] Osman Başkaya, Enis Sert, Volkan Cirik, and Deniz Yuret. Ai-ku: Using substitute vectors and co-occurrence modeling for word sense induction and disambiguation. In *Proc. of SemEval*, 2013.

[Biemann, 2012] Chris Biemann. Turk bootstrap word sense inventory 2.0: A large-scale resource for lexical substitution. In *LREC*, 2012.

[Bingel *et al.*, 2018] Joachim Bingel, Gustavo Paetzold, and Anders Søgaard. Lexi: A tool for adaptive, personalized text simplification. In *Proc. of COLING*, 2018.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, 2019.

[Erk *et al.*, 2009] Katrin Erk, Diana McCarthy, and Nicholas Gaylord. Investigations on word senses and word usages. In *Proc. of ACL*, 2009.

[Hou *et al.*, 2020] Bairu Hou, Fanchao Qi, Yuan Zang, Xurui Zhang, Zhiyuan Liu, and Maosong Sun. Try to substitute: An unsupervised chinese word sense disambiguation method based on hownet. In *Proc. of COLING*, 2020.

[Ide *et al.*, 2008] Nancy Ide, Collin Baker, Christiane Fellbaum, Charles Fillmore, and Rebecca Jane Passonneau. Masc: The manually annotated sub-corpus of american english. In *LREC*, 2008.

[Jia *et al.*, 2019] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. In *Proc. of EMNLP*, 2019.

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

[Kremer *et al.*, 2014] Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. What substitutes tell us-analysis of an "all-words" lexical substitution corpus. In *Proc. of EACL*, 2014.

[Liu *et al.*, 2019] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[McCarthy and Navigli, 2007] Diana McCarthy and Roberto Navigli. SemEval-2007 task 10: English lexical substitution task. In *Proc. of SemEval*, 2007.

[McCarthy and Navigli, 2009] Diana McCarthy and Roberto Navigli. The English lexical substitution task. *LRE*, 2009.

[Melamud *et al.*, 2015] Oren Melamud, Omer Levy, and Ido Dagan. A simple word embedding model for lexical substitution. In *Proc. of the 1st Workshop on Vector Space Modeling for NLP*, 2015.

[Melamud *et al.*, 2016] Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proc. of CoNLL*, 2016.

[Miller, 1995] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

[Reimers and Gurevych, 2019] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. of EMNLP*, 2019.

[Rousseeuw, 1987] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20, 1987.

[Sharoff, 2006] Serge Sharoff. Open-source corpora: Using the net to fish for linguistic data. *International journal of corpus linguistics*, 11(4), 2006.

[Soler *et al.*, 2019] Aina Garí Soler, Anne Cocos, Marianna Apidianaki, and Chris Callison-Burch. A comparison of context-sensitive models for lexical substitution. In *Proc. of IWCS*, 2019.

[Speer and Lowry-Duda, 2017] Robyn Speer and Joanna Lowry-Duda. ConceptNet at SemEval-2017 task 2: Extending word embeddings with multilingual relational knowledge. In *Proc. of SemEval*, 2017.

[Szarvas *et al.*, 2013a] György Szarvas, Chris Biemann, and Iryna Gurevych. Supervised all-words lexical substitution using delexicalized features. In *Proc. of NAACL*, 2013.

[Szarvas *et al.*, 2013b] György Szarvas, Róbert Busa-Fekete, and Eyke Hüllermeier. Learning to rank lexical substitutions. In *Proc. EMNLP*, 2013.

[Yang *et al.*, 2019] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances of NeurIPS*, 2019.

[Zhou *et al.*, 2019] Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. BERT-based lexical substitution. In *Proc. of ACL*, 2019.