On Refining BERT Contextualized Embeddings using Semantic Lexicons

Georgios Zervakis, Emmanuel Vincent, Miguel Couceiro, Marc Schoenauer

On Refining BERT Contextualized Embeddings using Semantic Lexicons

Georgios Zervakis[1][0000−0002−3015−2238], Emmanuel Vincent[1][0000−0002−0183−7289], Miguel Couceiro[1][0000−0003−2316−7623], and Marc Schoenauer[2][0000−0003−1450−6830]

[1] Université de Lorraine,CNRS, INRIA, LORIA, F-54000 Nancy, France
[2] INRIA TAU, LRI, France
{georgios.zervakis,emmanuel.vincent,miguel.couceiro,marc.schoenauer}@inria.fr

Abstract. Word vector representations play a fundamental role in many NLP applications. Exploiting human-curated knowledge was proven to improve the quality of word embeddings and their performance on many downstream tasks. Retrofitting is a simple and popular technique for refining distributional word embeddings based on relations coming from a semantic lexicon. Inspired by this technique, we present two methods for incorporating knowledge into contextualized embeddings. We evaluate these methods with BERT embeddings on three biomedical datasets for relation extraction and one movie review dataset for sentiment analysis. We demonstrate that the retrofitted vectors do not substantially impact the performance for these tasks, and conduct a qualitative analysis to provide further insights on this negative result.

Keywords: Contextualized embeddings · BERT · Knowledge integration · Retrofitting · Qualitative analysis

1 Introduction

The introduction of word embeddings was a breakthrough in NLP. Early approaches based on the *distributional hypothesis* — words that appear in the same context tend to be semantically similar — such as word2vec [11] provided a fixed embedding for each word. Recently, *contextualized embedding* systems like BERT [3] allow the generation of context-dependent word representations, which substantially improve the performance on many downstream NLP tasks.

Although such systems can be trained on data specific to the domain of interest, it is not yet clear how we can encode factual knowledge or impose constraints in the embeddings. Knowledge bases typically provide this type of information, hence it is reasonable to exploit them in order to obtain more accurate and explainable embeddings.

Retrofitting [4] is a popular technique that modifies any set of pretrained distributional word embeddings to account for relational information encoded by a semantic lexicon. This is done as a post-processing step using an iterative update method called belief propagation [1] on a graph of relations obtained

from the lexicon to update the word vectors. This method was proven to improve performance on various intrinsic and extrinsic evaluation tasks [2, 5, 9, 12, 13].

In this paper, we aim to extend retrofitting to operate with contextualized word embeddings. More specifically, we propose two different methods that, as in the original retrofitting approach, make use of similarity relations between words in order to move the respective embeddings closer to each other in the latent space. The first method combines the embedding of a given test sentence with the embeddings of sentences involving similar words in the training set, while the second method replaces a word in the test sentence by all possible similar words and combines the resulting embeddings. We evaluate the proposed methods with BERT embeddings on three biomedical datasets for a relation extraction task and one movie review dataset for sentiment analysis, and compare them with an oracle topline and two baselines (weighted majority vote and class posterior averaging). We show that both methods do not substantially impact the performance for this task, and conduct a qualitative analysis to provide further insights on this negative result.

The paper is organised as follows. We discuss related work in Section 2, and present the proposed methods in Section 3. We describe the experimental evaluation setup in Section 4, and we analyze the obtained results in Section 5. We provide conclusions and discuss future work in Section 6.

2 Related Work

There have been several attempts to improve the quality of word embeddings by incorporating knowledge into the process. Two main categories of methods can be distinguished, which we refer to as *joint* or *post-hoc*.

Joint methods integrate knowledge by retraining the embedding model from scratch using a modified training objective. For example, [10] proposed to replace the classical bag-of-words contexts in the word2vec Skip Gram model by dependency-based contexts, and showed that the resulting embeddings better reflect the syntactic similarities between words. In another approach, [19] modified a BiLSTM recurrent neural network to take into account information coming from the WordNet and NELL knowledge bases. To this end, they employed an attention mechanism that computes the relevance of candidate concepts from the knowledge base to the current input, and a second component that decides whether to exploit this information or not, and they reported improvements on both entity and event extraction tasks. In the same fashion, KnowBERT [15] incorporates WordNet and part of Wikipedia into BERT, showing the ability of the model to recall facts from the databases, improving downstream relation extraction, entity typing and word sense disambiguation tasks at the same time. Nonetheless, joint methods come with the downside that they are model-specific, and often time-consuming since they require retraining the system afresh.

Post-hoc methods surpass these limitations, since knowledge is inserted in the word embeddings after training, regardless of the model used to obtain them. The most popular technique among these is retrofitting [4]. This is a graph-based

approach that, given a semantic lexicon, i.e., a knowledge graph whose nodes represent words and edges represent relations between them, tries to reposition the word embeddings in such a way that they become closer (under some distance metric) to neighborhood embeddings in the graph. Initially, [4] considered a single type of relation between words, namely 'similarity'. Later approaches have extended retrofitting to account for 'dissimilarity' relations [9, 12, 13] and ordering (ranking) between the relations [6].

By default, all of the above retrofitting methods can only be applied to distributional word embeddings, i.e., a single representation vector per word. When we shift to contextualized embeddings, each word in the vocabulary can have a different representation in each sentence. An attempt to retrofit contextualized embeddings coming from ELMo is presented in the Paraphrase-aware Retrofitting (PAR) [16] method. More specifically, PAR learns an orthogonal transformation matrix that pulls closer the embeddings of words in paraphrased contexts, and separates those in unrelated contexts. However, this approach is limited to pairs of paraphrased contexts and cannot benefit from different sources of linguistic information. To our knowledge, there is no existing method for contextualized embeddings that takes full advantage of the benefits of retrofitting.

3 Proposed Contextualized Embedding Refinement Methods

As in the conventional retrofitting approaches discussed in Section 2, we assume a vocabulary of words $\mathcal{V} = \{w_1, \ldots, w_n\}$ and an ontology Ω of semantic relations between words in $\mathcal{V}$. We can then represent Ω in the form of an undirected graph $(\mathcal{V}, \mathcal{E})$, where nodes correspond to words in $\mathcal{V}$ and edges $(w_i, w_j) \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ to semantic relations between nodes. Now, suppose that we have a contextualized word representation model $\mathcal{M}$, along with a training corpus $\mathcal{D}_{\text{train}}$ on which it is fine-tuned and a test corpus $\mathcal{D}_{\text{test}}$ on which it is evaluated for a particular task.

3.1 Method A

The first proposed embedding refinement method, which we refer to as Method A, combines the contextualized embedding of a given word in the test set with the contextualized embeddings of all occurrences of all similar words in the training set. Let $\bar{q}_i \in \mathbb{R}^d$ be the contextualized embedding of word $w_i \in \mathcal{V}$ coming from $\mathcal{M}$ for a given test instance[3]. Let us further denote by $\mathcal{J}_i$ the set of words w_j which are adjacent to w_i according to Ω, and by $\mathcal{K}_j$ the set of training instances where w_j occurs. Then we define $\hat{q}_{jk} \in \mathbb{R}^d$ to be the contextualized embedding computed for all occurrences of w_j in $\mathcal{D}_{\text{train}}$, as index by $k \in \mathcal{K}_j$. The index sets $\mathcal{J}_i$ and $\mathcal{K}_j$ vary dynamically for every word.

[3] For simplicity, $\bar{q}_i$ does not have a superscript for the test sentence as we only process one test sentence at a time.

The goal is to learn a new embedding q_i that it is close to $\bar{q}_i$ and to adjacent nodes in Ω under the $\mathcal{L}_2$ norm by minimizing

$$\mathcal{L}(q_i) = \|q_i - \bar{q}_i\|^2 + \sum_{j \in \mathcal{J}_i} \sum_{k \in \mathcal{K}_j} b_{ijk} \|q_i - \hat{q}_{jk}\|^2 \tag{1}$$

The weights b_{ijk} must naturally depend on the number of neighbours $|\mathcal{J}_i|$ of w_i, and on the number of occurrences $|\mathcal{K}_j|$ of each neighbor w_j in $\mathcal{D}_{\text{train}}$. In the following we define them as $b_{ijk} = c_{ij} \times d_{jk} = \frac{1}{|\mathcal{J}_i|^\alpha} \cdot \frac{1}{|\mathcal{K}_j|^\beta}$, $\alpha, \beta \in [0, \infty)$ where c_{ij} controls the contribution of each neighbour and d_{jk} controls the contribution of each of its occurrences. For example, $\alpha = \beta = 0$ results in equal weights $b_{ijk} = 1$ for all occurrences, while $\alpha = \beta = 1$ results in weights b_{ijk} that sum up to 1.

Equating to zero the derivative of $\mathcal{L}$ with respect to q_i and expressing the $\sum_k b_{ijk} \hat{q}_{jk}$ in terms of the mean $\mu_{\hat{q}_j}$ of all $\hat{q}_{jk}$ results in the following update rule:

$$q_i = \frac{\bar{q}_i + \sum_j \sum_k b_{ijk} \hat{q}_{jk}}{1 + \sum_j \sum_k b_{ijk}} = \frac{\bar{q}_i + |\mathcal{J}_i|^{-\alpha} \sum_j |\mathcal{K}_j|^{1-\beta} \mu_{\hat{q}_j}}{1 + |\mathcal{J}_i|^{-\alpha} \sum_j \mathcal{K}_j^{1-\beta}}. \tag{2}$$

The retrofitting operation therefore takes the form of a weighted average of the original embedding and the embeddings of all occurrences of all similar words in the training set.

3.2 Method B

The second proposed method, which we refer to as Method B, does not involve $\mathcal{D}_{\text{train}}$ at all. Instead, everything happens at test time. Again, we utilise $\mathcal{M}$ to obtain the embedding $\bar{q}_i$ of word w_i for a specific sentence in $\mathcal{D}_{\text{test}}$. In addition, we derive one embedding $\hat{q}_j$ for every word w_j which is adjacent to w_i according to Ω. To do so, we create a new sentence by replacing w_i with w_j in the test sentence, and repeat for every adjacent node of w_i in Ω. The objective is once more to learn a new vector q_i that is close to both $\bar{q}_i$ and all $\hat{q}_j$ under the $\mathcal{L}_2$ norm by minimizing

$$\mathcal{L}(q_i) = \|q_i - \bar{q}_i\|^2 + \sum_{j \in \mathcal{J}_i} b_{ij} \|q_i - \hat{q}_j\|^2 \tag{3}$$

Similarly to the above, we define the weights as $b_{ij} = \frac{1}{|\mathcal{J}_i|^\alpha}$, $\alpha \in [0, \infty)$.

Equating to zero the derivative of $\mathcal{L}$ with respect to q_i and expressing the $\sum_j b_{ij} \hat{q}_j$ in terms of the mean $\mu_{\hat{q}_j}$ of all $\hat{q}_j$ results in the following update rule:

$$q_i = \frac{\bar{q}_i + \sum_j b_{ij} \hat{q}_j}{1 + \sum_j b_{ij}} = \frac{\bar{q}_i + |\mathcal{J}_i|^{1-\alpha} \mu_{\hat{q}_j}}{1 + |\mathcal{J}_i|^{1-\alpha}}. \tag{4}$$

Again, the retrofitting operation takes the form of a weighted average of the original embedding and the embeddings of all neighbouring words.

The main difference between the two methods lies in the way we exploit the information coming from the knowledge graph. Method A typically results in a

large number of neighbouring vectors $\hat{q}_{ik}$ that contain noise, since the context around the corresponding words differs from that of the test sentence in general. In contrast, Method B generates fewer neighbouring vectors $\hat{q}_j$ that share exactly the same context as the test sentence being processed.

4 Experimental Setup

In this section, we first provide information with respect to the data, the semantic lexicons and the contextual word embedding model we used to evaluate the proposed retrofitting methods. Then, we describe the experimental evaluation and we suggest three alternative strategies for comparison.

4.1 Data

We consider two tasks: relation extraction from biomedical data[4] and sentiment analysis of movie reviews. Two semantic verb lexicons are introduced in [2], referred to as **annotated** and **expanded clusters**. The former contains 192 verbs that appear frequently in a corpus of 2,230 biomedical journal articles, while the latter is an extended version of 1,149 verbs. Both lexicon come with three levels of granularity, i.e., verbs are grouped into 16, 34 and 50 classes[5], and are used for relation extraction.

ChemProt is a manually annotated corpus of relations between drugs/ chemical compounds and genes/proteins mentions found in PubMed abstracts. The relations are categorized into ten classes from which only five are used during evaluation. The task is to predict whether a pair of such entities is related or not, and if so, output the type of relation.

The **DDI** corpus aims in the development of systems that can automatically detect drug entities and drug-to-drug interactions in biomedical text. The corpus itself consists of texts from the DrugBank database and abstracts from the MedLine database. Annotations were provided by domain experts that classified drug-drug interactions into four DDI types.

i2b2 2010 corpus promotes the study of extraction/classification/relations of medical problems, tests, and treatments. The data consist of discharge summaries collected from Partners Healthcare, Beth Israel Deaconess Medical Center, and the University of Pittsburgh Medical Center, where relations of medical problems-treatments were grouped into eight classes.

For the sentiment analysis task, we use the exact same semantic lexicons as in [4], namely, **FrameNet**, **PPDB** and two variants of **WordNet** which we refer to as $\textbf{WordNet}_{\text{syn}}$ and $\textbf{WordNet}_{\text{all}}$ (see more details in [4]). The size of these lexicons is relatively large, since they are general and contain knowledge

[4] The biomedical datasets are included in the Biomedical Language Understanding Evaluation (BLUE) benchmark, as well as the preprocessing codes for creating the training, development and test sets.

[5] We refer to each different version of the verb lexicons simply by adding the number of the verb classes next to its name, e.g., annotated-34.

about words which do not convey any sentiment, e.g., pronouns, prepositions, etc.. In order to focus on relevant words for the task, in conjunction with the semantic lexicons we utilize the **Bing Liu Sentiment Lexicon** [7], a domain-independent list of 6,786 adjectives that is manually created and that categorizes words as either positive or negative according to their sentiment.

SST-2 (Stanford Sentiment Treebank) [17] is a collection of 11,855 sentences from movie reviews including human annotations of their sentiment. The goal is to classify a given sentence as either positive or negative. Since the test labels are not publicly available, we split the training set such that 13% of the sentences are used for testing and the remaining are used for training. The resulting test set has 462 positive and 438 negative reviews, while the training set has 3,148 positive and 2,872 negative reviews. Finally, we use the development set provided by the authors.

4.2 BERT Architecture and Retrofitting

There are different locations within the architecture of BERT, where retrofitting transformations can be applied. In general, the model consists of 12 Transformer blocks [18] followed by a pooling layer, i.e., a fully connected layer with a dropout layer and a *tanh* activation. Each block contains a sequence of transformations that is divided into layers. The output layer of each block consists of a linear transformation, followed by dropout and layer normalisation. For both approaches we experimented with four retrofitting different settings: before **or** after layer normalisation at Transformer block 11 **or** 12.

The motivation behind these choices is related to the complex architecture of the model. We hypothesize that the impact of any change into the embeddings would be more noticeable as we get closer to the output space, rather than in earlier layers of the model. Thus, we started experimenting at the pooling layer, which is the closest to the output space, but the results were not promising. Consequently, we moved one step back at the output layer of the last Transformer block, and further back to the same place of the preceding Transformer block.

In the retrofitting equations (1) or (3), we initially considered as $\bar{q}_i$ the embedding corresponding to the word token in the test sentence, but preliminary experiments showed that this did not have an impact on the final performance. To verify this, we replaced the embeddings of these individual words with random numbers, or even zeroes. Both cases did not affect the performance, indicating that the output classifier is not very much dependent on single word embeddings. Instead, we focus on the `[CLS]` token embedding which is a weighted linear average of all word embeddings in the test sentence, it is closer to the output space, and has a bigger impact on the final result. All $\hat{q}_{ij}$ in (1) correspond to the activations of the word token in training sentences, whereas all $\hat{q}_j$ in (3) correspond to the activations of the `[CLS]` token in modified test sentences.

4.3 Technical Details

For the relation extraction task we chose BlueBERT [14] a specific variant of BERT that is further pre-trained on PubMed abstracts and clinical notes from MIMIC-III database, while for sentiment analysis we experimented with the classical BERT. In particular, for both tasks we selected the BERT-Base release of the model, which makes use of the exact same configurations, (e.g., vocabulary, length) as in the original BERT, and we further fine-tuned it on the downstream task for each dataset. We treat both tasks as a sentence classification problem. For relation extraction the named entities are anonymized with pre-defined tags (e.g., `@GENE`, `@CHEMICAL` for ChemProt) as in [8]. Then, we feed an input sentence into BERT which makes use of the `[CLS]` token of that sentence to perform the classification. In particular, the `[CLS]` representation is forwarded into the output layer of the last Transformer block, that produces an estimation for each class.

4.4 Grid Search Optimization

In order to find a good set of values for the retrofitting hyperparameters α, β, we performed a grid search using the development sets. For the first approach, we used both annotated and expanded clusters and we searched for $\alpha, \beta \in [0, 2]$ with a step of 0.2. We do not proceed on testing Method A for SST-2, as it turns out to be inferior to Method B. For the second approach, we use all four lexicons for sentiment analysis in conjunction with Bing Liu's sentiment lexicon (explained in Section in 4.1), while for relation extraction we only used the 34 and 50 classes of the annotated clusters[6]. Once again, we performed a grid search on the development sets where we searched for $\alpha \in [0, 2]$ with a step of 0.2.

4.5 Alternative Classification Strategies

In order to assess the ability of our method to leverage the information in the lexicons, we augmented all datasets by adding all modified sentences that occur by replacing the underlying word with a neighbouring one, and compared with the following alternative strategies:

Topline: Always selecting the true class of a test sentence as the final prediction, if it was predicted by at least one of the original or the modified sentences.

Weighted majority vote (WMJ): Picking the predicted class with the most occurrences as the final prediction out of the original and the modified test sentences. Here, we assigned a weight of 1 to the original and a weight of $\frac{1}{|S|^\delta}, \delta \in [0, 1]$ to each modified sentence, where $|S|$ is the total number of sentences for the current test input. We experimentally noticed that choices of δ outside $[0, 1]$ did not affect the final prediction.

Average probabilities (AVGP): Averaging the probabilities of the predicted classes for both the original and the modified test sentences, and taking the class with the maximum probability as the final prediction.

[6] This is due to the extensive amount of neighbouring verbs on the annotated-16 and the expanded clusters, which significantly increases the computational cost.

5 Results and Qualitative Study

In this section we present the results obtained from the grid search, and conduct additional experiments that give more insight on the reasons why the proposed methods yield a similar performance to the baseline model.

5.1 Grid Search Experimental Results

After finding the best performing set of hyperparameters amongst all combinations of lexicons, Transformer blocks, and positions that were tested on the development set, we evaluated the corresponding model on the test set. We report the performance for each dataset in terms of micro F_1-score for relation extraction, and accuracy for sentiment analysis[7]. The results for both retrofitting approaches are displayed in Table 1. At first sight, both approaches seem to have no significant impact compared to the baseline performance. More specifically, Method A results in a decrease of performance on all datasets, while Method B slightly improves it for ChemProt and SST-2. Furthermore, we notice that in many cases the alternative strategies we propose work better than our retrofitting approaches. This suggests that i) the use of the lexicons is meaningful, but ii) we have not yet found the correct way of exploiting this knowledge. It is also worth highlighting the abrupt decrease in test performance on the i2b2-2010 for the AVGP method. We assume this is due to the model outputting different probabilities for each of the modified sentences. To confirm this, we compared with the score obtained from WMV for every $\delta \in [0, 1]$ with a step of 0.1, and we observed that for low values of the weight the performance is significantly worse. This indicates that the original sentence is more important than the modified ones, implying in turn that we should assign a higher weight on it. However, in AVGP the averaging equally favours each class, and thus performs poorly.

5.2 Euclidean Distance Ranking of Retrofitted Vectors

In order to understand in greater depth how our proposed methods change the embeddings in space, let us focus on a single test case[8] where the proportion of disagreements between the baseline model and the test case model is statistically significant (based on McNemar's test). This points out that both models behave differently, but on average they result in similar performance. To further analyse how Method A affects the embeddings in the latent space, we randomly select $5,000$ (out of $18,014$) test sentences where we apply our method, and we compute the corresponding activation of the `[CLS]` token before and after retrofitting. Next, we compute the Euclidian distance between every retrofitted vector and every `[CLS]` vector before retrofitting. This results in a 5000×5000 matrix, where

[7] This is the standard choice of metrics for these tasks and datasets [14, 17].

[8] This corresponds to Method A on ChemProt, using the expanded-16 clusters, and retrofitting after layer normalisation at Transformer block 12, with $\alpha = 0.4$ and $\beta = 1.4$ (second row of Table 1).

Table 1: Performance results across all datasets and proposed strategies as well as some retrofitting approaches for static word embeddings. Baseline corresponds to BERT base model finetuned on each dataset for the specific task. Method A, B denote the proposed retrofitting approaches. Topline, AVGP and WMV were discussed in Section 4.5, where for the last we select the weight (δ) based on the best performance on the validation set.

Corpus	**Model**	**Lexicon**	**Dev** miF_1/Acc	**Test** miF_1/Acc
ChemProt	Baseline	–	74.47	72.61
	Method A	expanded-16	74.86	72.56
	Method B	annotated-50	74.59	72.63
	Topline	annotated-50	75.54	73.67
	AVGP	annotated-50	72.92	72.07
	WMV ($\delta = 1.0$)	annotated-50	74.47	72.61
	Chiu et al. [2]	expanded-34	–	71.00
DDI	Baseline	–	71.34	80.11
	Method A	expanded-34	79.35	78.78
	Method B	annotated-34	72.33	79.43
	Topline	annotated-34	73.04	80.97
	AVGP	annotated-34	71.97	79.40
	WMV ($\delta = 0.1$)	annotated-34	72.02	79.60
i2b2-2010	Baseline	–	71.34	72.69
	Method A	expanded-16	72.92	72.52
	Method B	annotated-34	71.83	72.63
	Topline	annotated-34	73.71	74.18
	AVGP	annotated-34	60.79	58.50
	WMV ($\delta = 1.0$)	annotated-34	71.34	72.69
SST-2	Baseline	–	91.86	92.00
	Method B	$\text{WordNet}_{\text{syn}}$	92.09	92.11
	Topline	$\text{WordNet}_{\text{syn}}$	94.95	94.55
	AVGP	$\text{WordNet}_{\text{syn}}$	90.37	90.11
	WMV ($\delta = 1.0$)	$\text{WordNet}_{\text{syn}}$	91.86	92.00
	Faruqui et al. [4]	$\text{WordNet}_{\text{syn}}$	–	82.40

each row contains the distances of one retrofitted vector to all original vectors (before retrofitting). We then rank from $0 - 5000$ each retrofitted embedding by sorting each row in the matrix in ascending order. By doing so, we can check how far our method is moving the embeddings in the latent space. The distribution of the resulting rankings across all vectors is summarized in the histogram in Figure 1. From this plot, we can observe that a large proportion of vectors has a relatively low ranking (around $[0, 80]$), but there is also a considerable amount of vectors with high ranking (around $[950, 1000]$), suggesting that potentially the vectors do not move as far as they should, or sometimes they move too far. This is an indication that there is a lot of variation in the neighbouring embeddings, and therefore not all words in the lexicons are relevant for the task at hand. The following experiment will check if restricting the lexicons to the domain has any impact when retrofitting.

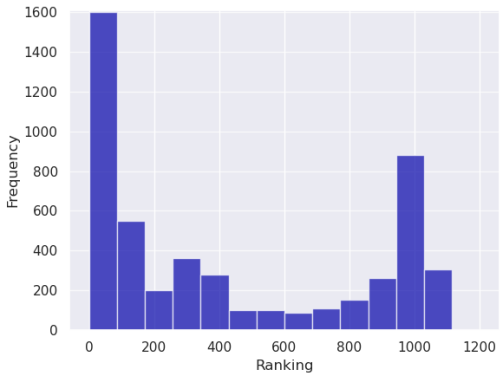

Fig. 1: Histogram of the ranking across `[CLS]` token retrofitted vectors for all 5000 ChemProt test sentences where Method A is applied.

5.3 Neighbouring Word Filtering

Bing Liu's list of adjectives allow us to focus on appropriate words in the semantic lexicons for the task of sentiment analysis. The next question we want to answer is which neighbouring words are relevant for the underlying word, and which are not. It is evident that not all neighbouring words for a given word in the lexicons are actual synonyms in the context of movie reviews. Replacing single words in the input sentence in Method B, forces the same context between the original and the modified sentence. Consequently, we restrict the lexicons to the domain by selecting neighbours that are "good" replacements instead of using the whole list. This is done by inspecting the predictions of BERT for every original and modified sentence on the augmented development set for a given lexicon (see Section 4.5). Then, we can distinguish between the following cases: **(A)** the original sentence was wrongly classified but the modified sentence was correctly classified (good case), **(B)** the original and the modified sentence were correctly/wrongly classified (neutral case), and **(C)** the original sentence was correctly classified but the modified sentence was wrongly classified (bad case).

Next, we compute the counts that correspond to good, neutral and bad cases for every pair of original-neighbouring word. These will show on average if a neighbour is a good replacement or not for a given word. Then, using the McNemar's statistical test, we create three reduced versions, one for each semantic lexicon, by selecting a neighbour for a given word with a 10%, 50% and 90% confidence level[9]. The higher the confidence level the more certain we are about replacing a word by another one, but the smaller the lexicon becomes (and vice versa). Finally, we repeat the grid search optimisation (see Section 4.4) and present in Table 2 the results for the best settings.

[9] We use the confidence level percentage as a subscript to denote the reduced lexicon, e.g., $\text{FrameNet}_{90\%}$.

Table 2: Results for the best performing lexicons derived from our neighbouring word selection for Method B and the proposed alternative strategies. Baseline corresponds to BERT base model, fine-tuned on SST-2 for sentiment analysis.

Lexicon	**Model**	**Dev** *Acc*	**Test** *Acc*
–	Baseline	91.86	92.00
$\text{FrameNet}_{10\%}$	Method B	92.09	92.00
	Topline	92.09	92.11
	AVGP	92.09	92.00
	WMV ($\delta = 0$)	92.09	92.00
$\text{WordNet}_{syn_{10\%}}$	Method B	92.09	92.00
	Topline	92.66	92.00
	AVGP	92.09	91.89
	WMV ($\delta = 0$)	92.09	92.00

Overall, there is some gain in performance compared to the baseline on the development set which is expected. For example, Method B reaches Topline performance for $\text{FrameNet}_{10\%}$, which suggests that retrofitting in the sense of averaging embeddings can be meaningful. Moreover, we can see that the Topline performance is almost identical to that of the baseline model on the test data. This is due to the limited size of the reduced lexicons[10]. Ideally, if the dataset were bigger, we would have selected lexicons with higher confidence level that would also be large enough to improve over the baseline, i.e., the Topline score would significantly outperform the baseline.

6 Conclusion and Future Work

In this paper, we proposed two approaches that extend the original retrofitting technique to operate with contextualized embedding systems. More precisely, we incorporated external knowledge coming from semantic lexicons into BERT contextualized representations. After conducting a large-scale series of experiments on three biomedical datasets for relation extraction, and one movie review dataset for sentiment analysis, we observe that both approaches do not substantially affect the performance on these downstream tasks. Our test results show that the lexicons can be a useful source of information to further improve the results. However, the current experimental setting did not make it viable. This is demonstrated in our qualitative study, where we show that when we improve the quality of the semantic lexicons by selecting only relevant neighbours for a given word, the resulting lexicons are not sufficiently large to be able to generalize at test time. In the future, we plan to experiment with more fine-grained tasks where we are certain about the knowledge source, and where we would not need to heavily depend on word statistics to apply the proposed method.

[10] For example FrameNet originally consists of 1700 words and 90140 relations, while its largest reduced version, $\text{FrameNet}_{10\%}$, has only 1 word and 5 relations.

References

1. Bengio, Y., *et al.*: Label propagation and quadratic criterion. In: Semi-Supervised Learning. pp. 193–216. MIT Press (2006)
2. Chiu, B.*et al.*.: Enhancing biomedical word embeddings by retrofitting to verb clusters. In: BioNLP. pp. 125–134 (2019)
3. Devlin, J., *et al.*: BERT: Pre-training of deep bidirectional transformers for language understanding. In: HLT-NAACL. pp. 4171–4186 (2019)
4. Faruqui, M., *et al.*: Retrofitting word vectors to semantic lexicons. In: NAACL HLT. pp. 1606–1615 (2015)
5. Ferret, O.: Turning distributional thesauri into word vectors for synonym extraction and expansion. In: IJCNLP. pp. 273–283 (2017)
6. Ferret, O.: Turning distributional thesauri into word vectors for synonym extraction and expansion (2017)
7. Hu, M., Liu, B.: Mining and summarizing customer reviews. pp. 168–177 (2004)
8. Lee, J., *et al.*: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
9. Lengerich, B., *et al.*: Retrofitting distributional embeddings to knowledge graphs with functional relations. In: COLING. pp. 2423–2436 (2018)
10. Levy, O., *et al.*: Dependency-based word embeddings. In: ACL. pp. 302–308 (2014)
11. Mikolov, T., *et al.*: Distributed representations of words and phrases and their compositionality. In: NIPS. pp. 3111–3119 (2013)
12. Mrkšić, N., *et al.*: Counter-fitting word vectors to linguistic constraints. arXiv preprint arXiv:1603.00892 (2016)
13. Mrkšić, N., *et al.*: Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. TACL **5**, 309–324 (2017)
14. Peng, Y., *et al.* biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In: BioNLP. pp. 58–65 (2019)
15. Peters, M.E.*et al.*.: Knowledge enhanced contextual word representations. arXiv preprint arXiv:1909.04164 (2019)
16. Shi, W., *et al.* contextualized word embeddings with paraphrases. In: EMNLP-IJCNLP. pp. 1198–1203 (2019)
17. Socher, R., *et al.*C.D., compositionality over a sentiment treebank. In: EMNLP. pp. 1631–1642 (2013)
18. Vaswani, A., *et al.*: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
19. Yang, B., *et al.*: Leveraging knowledge bases in LSTMs for improving machine reading. In: ACL. pp. 1436–1446 (2017)

Acknowledgements

This research was partially supported by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 952215 TAILOR and by the Inria Project Lab "Hybrid Approaches for Interpretable AI" (HyA-IAI). Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).