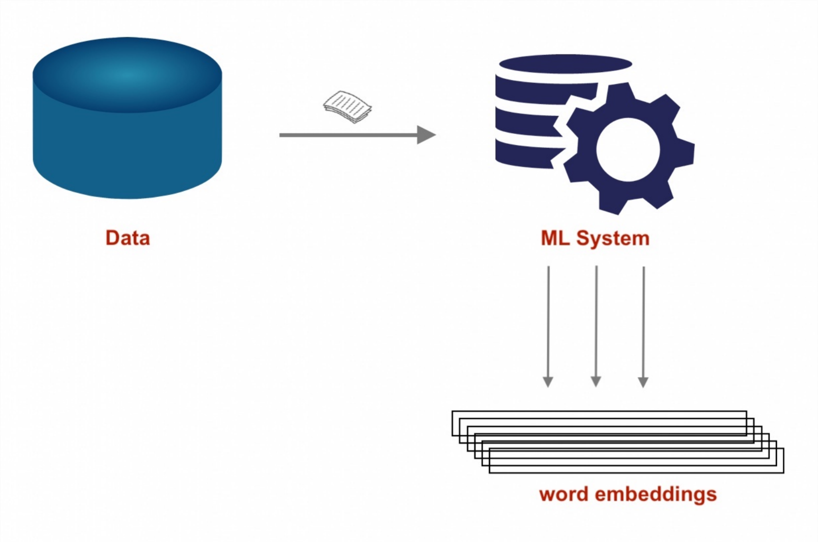
On Refining BERT Contextualized Embeddings using Semantic Lexicons*

Georgios Zervakis, Emmanuel Vincent, Miguel Couceiro, Marc Schoenauer

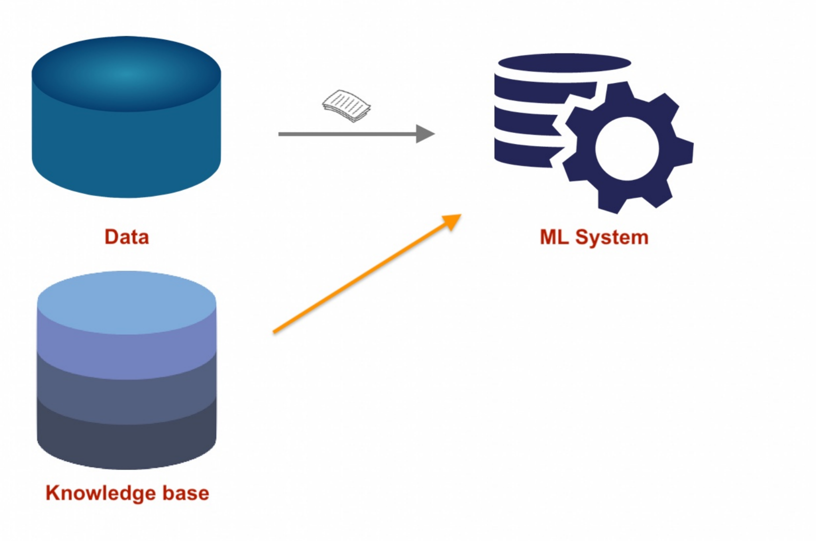
submitted to
ECML PKDD 2021

**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" (HyAIAI)*

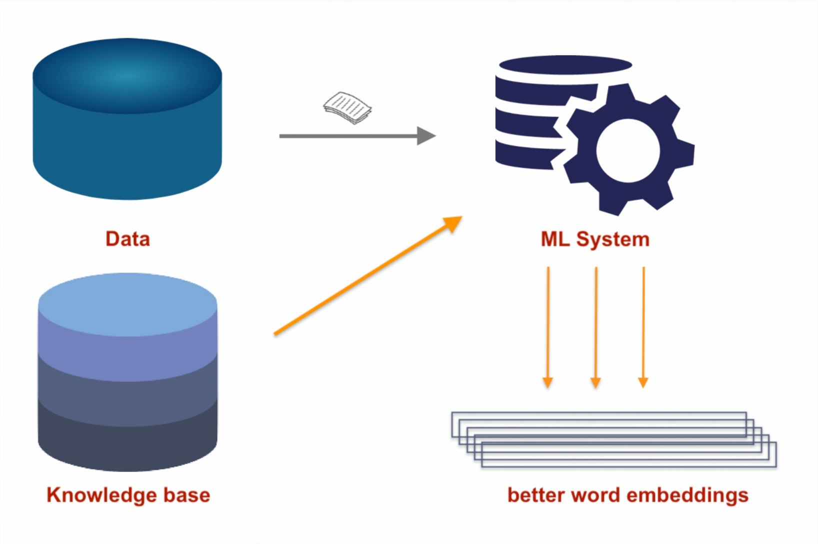
Challenge



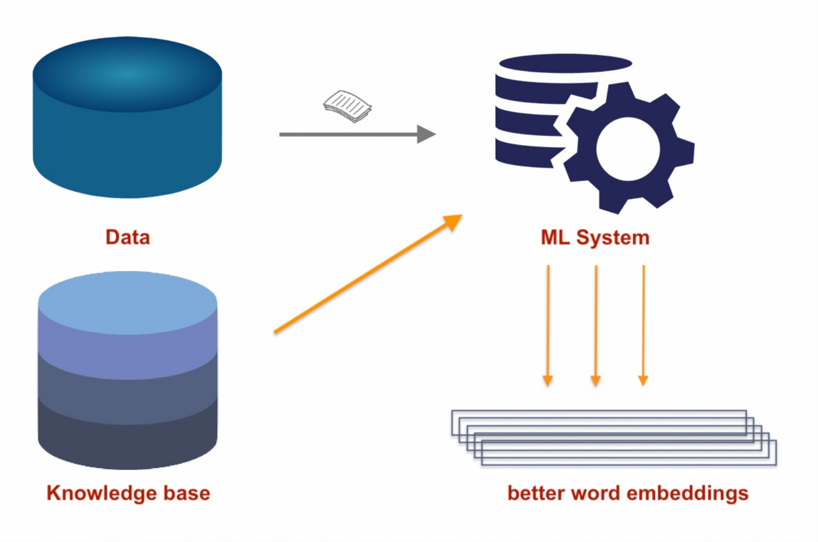
Challenge



Challenge



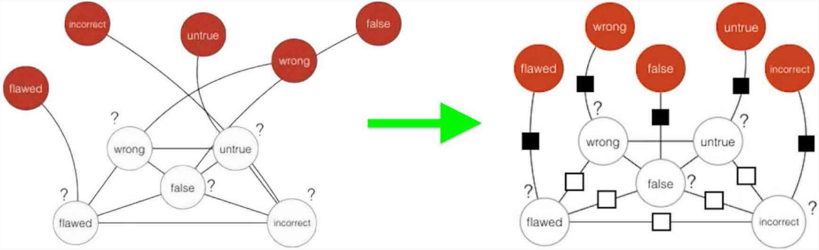
Challenge



Question: How can we encode external knowledge into word embeddings?

Retrofitting [1]

$V = \{w_i\}_1^n$: word vocabulary and Ω: ontology with semantic relations between words in $V \Rightarrow$ graph (V, E) where each vertex corresponds to a word-type and edges $(w_i, w_j) \in E \subseteq (V \times V)$ represent the relations
$Q' = (q'_1, \ldots, q'_n)$: matrix of learned static word embeddings $q'_i \in \mathbb{R}^d$

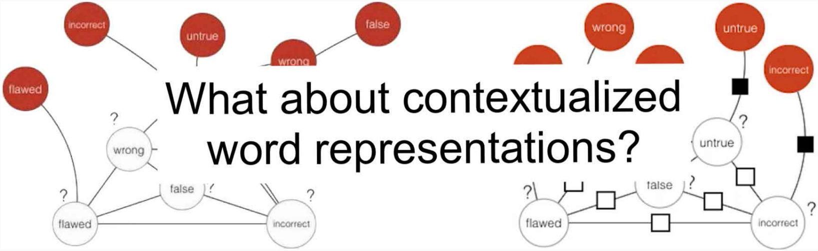


Objective: Learn a matrix $Q = (q_1, \ldots, q_n)$ s.t. q_i are close to q'_i and to adjacent vertices in Ω

$$\mathcal{L}(Q) = \sum_{i=1}^{n} \left[\alpha_i ||q_i - q'_i||^2 + \sum_{(i,j)\in E} \beta_{ij} ||q_i - q_j||^2 \right]$$

Retrofitting

$V = \{w_i\}_1^n$: word vocabulary and Ω: ontology with semantic relations between words in $V \Rightarrow$ graph (V, E) where each vertex corresponds to a word-type and edges $(w_i, w_j) \in E \subseteq (V \times V)$ represent the relations $Q' = (q'_1, \ldots, q'_n)$: matrix of learned static word embeddings $q'_i \in \mathbb{R}^d$


What about contextualized word representations?

Objective: Learn a matrix $Q = (q_1, \ldots, q_n)$ s.t. q_i are close to q'_i and to adjacent vertices in Ω

$$\mathcal{L}(Q) = \sum_{i=1}^{n} \left[\alpha_i ||q_i - q'_i||^2 + \sum_{(i,j) \in E} \beta_{ij} ||q_i - q_j||^2 \right]$$

Proposed Contextualized Embedding Refinement Methods

As in the conventional retrofitting we assume:

$V = \{w_i\}_1^n$: word vocabulary
Ω: ontology with semantic relations between words in V that is represented as a graph (V, E) where each vertex corresponds to a word-type and edges $(w_i, w_j) \in E \subseteq (V \times V)$ represent the relations

Furthermore:

$\mathcal{M}$: a contextualized word representation model
$\mathcal{D}_{\mathrm{train}}$: a training corpus on which $\mathcal{M}$ is fine-tuned for a particular task
$\mathcal{D}_{\mathrm{test}}$: a test corpus on which it is evaluated for this specific task

Method A

Idea: combine the contextualized embedding of a given word in $\mathcal{D}_{\mathrm{test}}$ with the contextualized embeddings of all occurrences of all similar words in $\mathcal{D}_{\mathrm{train}}$

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

$$b_{ijk} = c_{ij} \times d_{jk} = \frac{1}{|\mathcal{J}_i|^\alpha} \times \frac{1}{|\mathcal{K}_j|^\beta}, \quad \alpha, \beta \in [0, \infty)$$

- $\bar{q}_i$: the contextualized embedding for a word $w_i \in \mathcal{V}$ computed for a given sentence in $\mathcal{D}_{\mathrm{test}}$ using $\mathcal{M}$
- $\mathcal{J}_i$: the set of words w_j which are adjacent to w_i according to Ω
- $\hat{q}_{jk}$: the contextualized embeddings computed for all occurrences of w_j in $\mathcal{D}_{\mathrm{train}}$, as index by $k \in \mathcal{K}_j$
- c_{ij}, d_{jk} control the contribution of each neighbour and each of its occurrences respectively

Update Rules

Equating to zero the derivative of $\mathcal{L}$ with respect to q_i results in the following update rule:

$$q_i = \frac{\bar{q}_i + \sum_j \sum_k b_{ijk} \hat{q}_{jk}}{1 + \sum_j \sum_k b_{ijk}}$$

or, equivalently, by expressing $\sum_k b_{ijk} \hat{q}_{jk}$ in terms of the mean $\mu_{\hat{\mathbf{q}}_j}$ of all $\hat{q}_{jk}$ in the above equation:

$$q_i = \frac{\bar{q}_i + |\mathcal{J}_i|^{-\alpha} \sum_j |\mathcal{K}_j|^{1-\beta} \mu_{\hat{\mathbf{q}}_j}}{1 + |\mathcal{J}_i|^{-\alpha} \sum_j \mathcal{K}_j^{1-\beta}}.$$

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

Method B

Idea: combine the contextualized embedding of a given word in $\mathcal{D}_{\text{test}}$ with the contextualized embeddings that occur each time by replacing that word in the test sentence with every adjacent word in Ω.

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

$$b_{ij} = \frac{1}{|\mathcal{J}_i|^\alpha}, \quad \alpha \in [0, \infty).$$

- $\bar{q}_i$: the contextualized embedding for a word $w_i \in \mathcal{V}$ computed for a given sentence in $\mathcal{D}_{\text{test}}$ using $\mathcal{M}$
- $\mathcal{J}_i$: the set of words w_j which are adjacent to w_i according to Ω
- $\hat{q}_j$: the contextualized embeddings for every word w_j which is adjacent to w_i according to Ω. To compute those we input $\mathcal{M}$ with a new sentence by replacing w_i with w_j in the original test sentence, and repeat for every neighbour w_j in Ω.
- b_{ij} controls the contribution of each neighbour

Update Rules

Equating to zero the derivative of $\mathcal{L}$ with respect to q_i results in the following update rule:

$$q_i = \frac{\bar{q}_i + \sum_j b_{ij}\hat{q}_j}{1 + \sum_j b_{ij}}$$

or, equivalently, by expressing $\sum_j b_{ij}\hat{q}_j$ in terms of the mean $\mu_{\hat{q}_j}$ of all $\hat{q}_j$ in the above equation:

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

Once again the retrofitting operation takes the form of a weighted average of the original embedding and the embeddings of similar words

Experimental Setup: Relation Extraction

ChemProt [2]: relations between drugs/chemical compounds and genes/proteins mentions found in PubMed abstracts

DDI-2013[1]: drug-to-drug interaction in biomedical texts from the DrugBank database and abstracts from MedLine database

i2b2-2010[2]: relations of medical problems-treatments collected from discharge summaries

Verb Lexicons [3]: clusters of verbs annotated by humans using a corpus of biomedical journal articles (annotated clusters) or further extended automatically with relevant verbs from PubMed abstacts/articles (expanded clusters)

BlueBERT [4]: a specific variant of BERT that is further pre-trained on PubMed abstracts and clinical notes from MIMIC-III database

[1] https://www.cs.york.ac.uk/semeval-2013/task9/

[2] https://academic.oup.com/jamia/article/18/5/552/830538

Experimental Setup: Sentiment Analysis

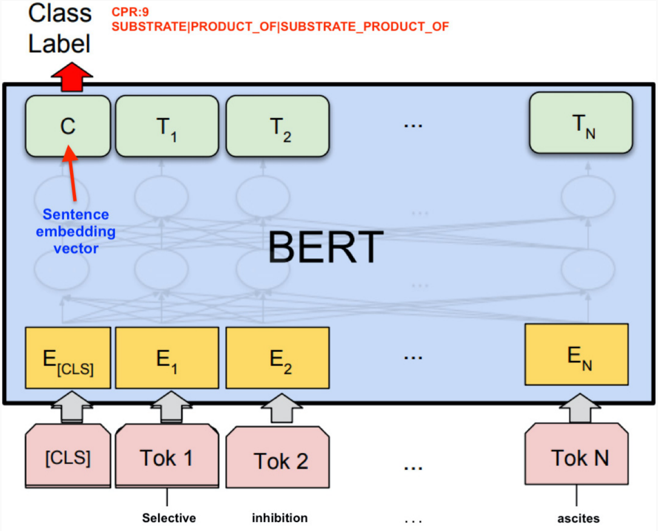
SST-2 [5]: collection of sentences from movie reviews including human-level annotations of their sentiment (either positive or negative).

Semantic Lexicons: FrameNet [6] PPDB [7] WordNet [8]

Bing Liu Sentiment Lexicon [9]: 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

BERT-Base [10]: the classical BERT model that is pre-trained on text from the BooksCorpus and the English Wikipedia

BERT for Sentence Classification: standard



Related Entities: prostaglandin, cyclooxygenase 2

Input Sentence: *Selective inhibition of prostaglandin spares renal function and cyclooxygenase 2 synthesis in cirrhotic rats with ascites*

BERT architecture and retrofitting

BERT consists of 12 Transformer blocks [11] followed by a pooling layer, i.e., 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 fully connected layer with a dropout and a layer normalisation [12]

For both methods we experimented with four different settings:

1. Retrofitting before layer normalisation at Transformer block 11
2. Retrofitting after layer normalisation at Transformer block 11
3. Retrofitting before layer normalisation at Transformer block 12
4. Retrofitting after layer normalisation at Transformer block 12

Alternative classification strategies

We can gain some insight by *augmenting* the datasets and comparing retrofitting with the following alternatives:

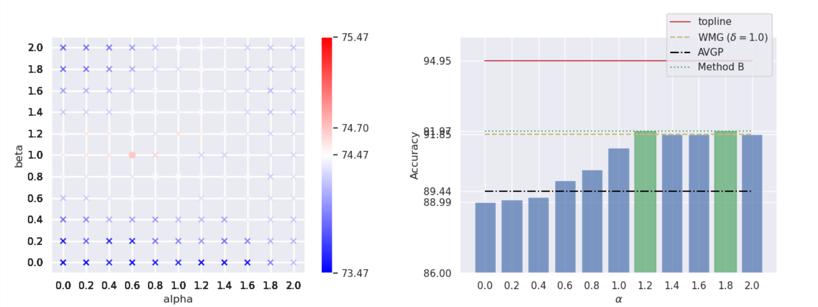
Topline: always selecting the true class of a test sentence as the final prediction, if it was predicted by at least one of the original or the modified

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 assign 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 of $[0, 1]$ 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.

Grid Search Optimisation

In order to find a good set of values for the retrofitting hyperparameters α, β, we performed a grid search using the development sets.



Left: Grid search plot of micro F_1-scores for Method A. The white colour corresponds to the baseline score while the red asterisk indicates the best (α, β)-pair performance on the dev set. **Right**: Grid search plot of accuracy scores for Method B. The green bars indicate the best α-values on the dev set, while the horizontal lines show the top performance of our proposed strategies.

Grid Search Results

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
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	WordNet$_{\mathrm{syn}}$	92.09	92.11
	Topline	WordNet$_{\mathrm{syn}}$	94.95	94.55
	AVGP	WordNet$_{\mathrm{syn}}$	90.37	90.11
	WMV ($\delta = 1.0$)	WordNet$_{\mathrm{syn}}$	91.86	92.00

Neighbouring Word Filtering

Question: Which neighbouring words are relevant for the underlying word, and which are not?

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.

Then, either:

i) the original sentence was wrongly classified but the modified sentence was correctly classified (good case)

ii) the original and the modified sentence were correctly/wrongly classified (neutral case)

iii) the original sentence was correctly classified but the modified sentence was wrongly classified (bad case)

Neighbouring Word Filtering

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.

For example, on PPDB semantic lexicon:

word pair	good	neutral	bad
(better, enhance)	2	10	0
(better, enhanced)	2	10	0
(better, best)	3	9	0
(better, brighter)	2	10	0

We create three reduced versions (one per semantic lexicon) by selecting a neighbour for a given word with a 10%, 50% and 90% confidence level (based on McNemar's test) and repeat the grid search experiment.

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).

Neighbouring Word Filtering Results

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

Gain in performance compared to the baseline on the development set as expected.

Topline performance for FrameNet$_{10\%}$ which suggests that retrofitting in the sense of averaging embeddings can be meaningful.

What about generalisation on the test data?

Neighbouring Word Filtering Results

Lexicon	# Words	# Edges	Lexicon	# Words	# Edges
FrameNet	1700	90140	FrameNet$_{10\%}$	1	5
PPDB	4893	44829	PPDB$_{10\%}$	1	6
WordNet$_{\mathrm{syn}}$	5481	29848	WordNet$_{syn_{10\%}}$	4	6
WordNet$_{\mathrm{all}}$	5481	113792	WordNet$_{all_{10\%}}$	6	9
FrameNet$_{50\%}$	–	–	FrameNet$_{90\%}$	–	–
PPDB$_{50\%}$	1	1	PPDB$_{90\%}$	–	–
WordNet$_{syn_{50\%}}$	2	2	WordNet$_{syn_{90\%}}$	1	1
WordNet$_{all_{50\%}}$	1	1	WordNet$_{all_{90\%}}$	–	–

Topline performance is almost identical to that of the baseline model on the test data.

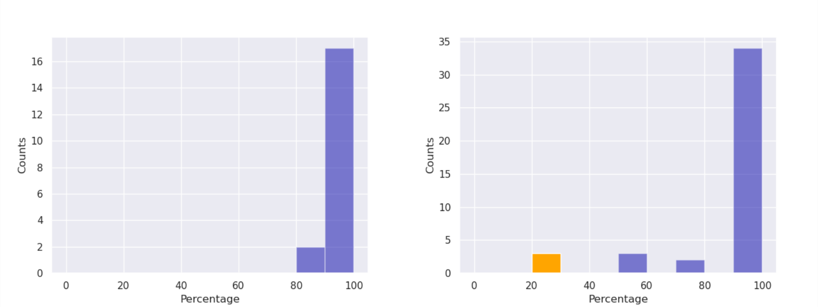
This is due to the limited size of the reduced lexicons.

If the dataset was 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 more significantly outperform the baseline.

How does Averaging Compare to Majority Vote?

Count how many times Method B yields the correct answer when the predictions of the modified sentences are 0-10% correct, up to 90-100% correct.

For example, we can see the distribution of these counts for Framenet$_{10\%}$ (right) and WordNet$_{syn_{10\%}}$ (left) on the dev set.



Averaging preserves the majority vote so there is hope in retrofitting provided the lexicon can help.

Conclusion and Future Work

We proposed two approaches that extend the original retrofitting technique to operate with BERT contextualized embedding.

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.

References

[1] Manaal Faruqui and *et al.*
Retrofitting word vectors to semantic lexicons.
In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615. Association for Computational Linguistics, 2015.

[2] Martin Krallinger and *et al.*
Overview of the biocreative vi chemical-protein interaction track.
In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, pages 141–146, 2017.

[3] Billy *et al.* Chiu.
Enhancing biomedical word embeddings by retrofitting to verb clusters.
In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 125–134, 2019.

[4] Yifan Peng, Shankai *et al.*
Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets.
In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, 2019.

[5] Richard Socher, Alex *et al.*Christopher D.
Recursive deep models for semantic compositionality over a sentiment treebank.
In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, October 2013.

[6] Collin F. Baker, Charles J. Fillmore, and John B. Lowe.
The Berkeley FrameNet project.
In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, 1998.

[7] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch.
PPDB: The paraphrase database.
In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, June 2013.

[8] George A. Miller.
Wordnet: A lexical database for english.
Commun. ACM, 38(11):39–41, 1995.

[9] Minqing Hu and Bing Liu.
Mining and summarizing customer reviews.
pages 168–177, 08 2004.

[10] Jacob Devlin and *et al.*
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, 2019.

[11] Ashish Vaswani and *et al.*
Attention is all you need.
In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[12] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton.
Layer normalization.
arXiv preprint arXiv:1607.06450, 2016.