Towards Better Context-aware Lexical Semantics: Adjusting Contextualized Representations through Static Anchors

Qianchu Liu, Diana McCarthy, Anna Korhonen
Language Technology Lab, TAL, University of Cambridge, UK
ql261@cam.ac.uk, diana@dianamccarthy.co.uk, alk23@cam.ac.uk

Abstract

One of the most powerful features of contextualized models is their dynamic embeddings for words in context, leading to state-of-the-art representations for context-aware lexical semantics. In this paper, we present a post-processing technique that enhances these representations by learning a transformation through static anchors. Our method requires only another pre-trained model and no labeled data is needed. We show consistent improvement in a range of benchmark tasks that test contextual variations of meaning both across different usages of a word and across different words as they are used in context. We demonstrate that while the original contextual representations can be improved by another embedding space from either contextualized or static models, the static embeddings, which have lower computational requirements, provide the most gains.

1 Introduction

Word representations are fundamental in Natural Language Processing (NLP) (Bengio et al., 2003). Recently, there has been a surge of contextualized models that achieve state-of-the-art in many NLP benchmark tasks (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019b; Yang et al., 2019). Even better performance has been reported from fine-tuning or training multiple contextualized models for a specific task such as question answering (Devlin et al., 2019; Xu et al., 2020). However, little has been explored on directly leveraging the many off-the-shelf pre-trained models to improve task-independent representations for lexical semantics. Furthermore, classic static embeddings are often overlooked in this trend towards contextualized models. As opposed to *contextualized* embeddings that generate dynamic representations for words in context, *static* embeddings such as word2vec (Mikolov et al., 2013) assign one fixed representation for each word. Despite being less effective in capturing context-sensitive word meanings, static embeddings still achieve better performance than contextualized embeddings in traditional context-independent lexical semantic tasks including word similarity and analogy (Wang et al., 2019). This suggests that static embeddings have the potential to offer complementary semantic information to enhance contextualized models for lexical semantics.

We bridge the aforementioned gaps and propose a general framework that improves contextualized representations by leveraging other pre-trained contextualized/static models. We achieve this by using *static anchors* (the average contextual representations for each word) to transform the original contextualized model, guided by the embedding space from another model. We assess the overall quality of a model's lexical semantic representation by two *Inter Word* tasks that measure relations between different words in context. We also evaluate on three *Within Word* tasks that test the contextual effect from different usages of the same word/word pair. Our method obtains consistent improvement across all these context-aware lexical semantic tasks. We demonstrate the particular strength of leveraging static embeddings, and offer insights on the reasons behind the improvement. Our method also has minimum computational complexity and requires no labeled data.

2 Background

This section briefly introduces the contextualized/static models that we experimented in this study. For static models, we select three representative methods. **SGNS** (Mikolov et al., 2013), as the most successful variant of word2vec, trains a log linear model to predict context words given a target word with negative sampling in the objective. **FastText** improves over SGNS by training at the n-gram level and can generalize to unseen words (Bojanowski et al., 2017). In addition to these two prediction-based models, we also include

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 4066–4075,
November 16–20, 2020. ©2020 Association for Computational Linguistics

one count-based model, **GloVe** (Pennington et al., 2014). GloVe is trained to encode semantic relations exhibited in the ratios of word-word occurrence probabilities into word vectors.

As opposed to static embeddings, contextualized models provide dynamic lexical representations as hidden layers in deep neural networks typically pre-trained with language modeling objectives. In our study, we choose three state-of-the-art contextualized models. **BERT** (Devlin et al., 2019) trains bidirectional transformers (Vaswani et al., 2017) with masked language modeling and next sentence prediction objectives. Liu et al. (2019b)'s **RoBERTa** further improves upon BERT by carefully optimizing a series of design decisions. **XLNet** (Yang et al., 2019) takes a generalized autoregressive pre-training approach and integrates ideas from Transformer-XL (Dai et al., 2019). For the best performance, we use the Large Cased[1] variant for each contextualized model. Since our study focuses on generic lexical representations and many of the lexical semantic tasks do not provide training data, we extract features[2] from these contextualized models without fine-tuning the weights for a specific task. This feature-based approach is also more efficient compared with fine-tuning the increasingly larger models which can have hundreds of millions of parameters.

3 Method

Our method[3] is built from a recently proposed cross-lingual alignment technique called *meeting in the middle* (Doval et al., 2018). Their method relies on manual translations to learn a transformation over an orthogonal alignment for better **cross-lingual static** embeddings. We show that by a similar alignment + transformation technique, we can improve **monolingual contextualized** embeddings without resorting to any labeled data.

The direct correspondence among contextualized and static embeddings for alignment is not straightforward, as contextualized models can compute infinite representations for infinite contexts. Inspired by previous study (Schuster et al., 2019) that found contextualized embeddings roughly form word clusters, we take the average of each word's contextual representations as anchors of a contextualized model. We call them *static anchors* as they provide one fixed representation per word, and therefore correspond to word embeddings from a static model such as FastText. We also use these anchors to align between contextualized models. To form the vocabulary for creating static anchors in our experiments, we take the top 200k most frequent words and extract their contexts from English Wikipedia.

To describe the method in more detail, we represent the anchor embeddings from the original contextualized model as our source matrix $\mathbf{S}$, and the corresponding representations from another contextualized/static model as target matrix $\mathbf{T}$. $\mathbf{s}_i$ and $\mathbf{t}_i$ are the source and target vectors for the i_{th} word in the vocabulary (V). We first find an orthogonal alignment matrix $\mathbf{W}$ that rotates the target space to the source space by solving the least squares linear regression problem in Eq. 1. $\mathbf{W}$ is found through Procrustes analysis (Schönemann, 1966).

$$\mathbf{W} = \arg\min_{\mathbf{W}} \sum_{i=1}^{|V|} \|\mathbf{W}\mathbf{t}_i - \mathbf{s}_i\|^2 \quad s.t. \quad \mathbf{W}^{\mathbf{T}}\mathbf{W} = \mathbf{I}. \tag{1}$$

As described in Eq. 2, we then learn a linear mapping $\mathbf{M}$ to transform the source space towards the average of source and the rotated target space, by minimizing the squared Euclidean distance between each transformed source vector $\mathbf{M}\mathbf{s}_i$ and the mean vector $\boldsymbol{\mu}_i$ ($\boldsymbol{\mu}_i = (\mathbf{s}_i + \mathbf{W}\mathbf{t}_i)/2$). $\mathbf{M}$ is the mapping we will use to transform the original contextualized space. Following Doval et al. (2018), $\mathbf{M}$ is found via a closed-form solution.

$$\mathbf{M} = \arg\min_{\mathbf{M}} \sum_{i}^{|V|} \|\mathbf{M}\mathbf{s}_i - \boldsymbol{\mu}_i\|^2 \tag{2}$$

For improved alignment quality, as advised by Artetxe et al. (2016), we normalize and mean-center[4] the embeddings in $\mathbf{S}$ and $\mathbf{T}$ a priori.

4 Experiments

Task Descriptions[5] We evaluate on three *Within Word* tasks. Usage Similarity (**Usim**) (Erk et al., 2013) dataset measures graded similarity of the same word in pairs of different contexts on the scale from 1 to 5. Word in Context (**WiC**) (Pilehvar and Camacho-Collados, 2019) dataset challenges a system to predict a binary choice of whether a pair of contexts for the same word belongs to the same

[1] 1024 dimensions with case-preserving vocabulary.

[2] Appendix A contains more details on feature extraction.

[3] Implementation details are listed in Appendix B.

[4] We pre-process representations with the same centering and normalization in all tasks. Our reported results are similar or better than the results from un-preprocessed representations.

[5] Appendix C reports details for each task and experiment.

meaning or not. We follow the advised training scheme in the original paper to learn a cosine similarity threshold on the representations. The recently proposed **CoSimlex** (Armendariz et al., 2019) task provides contexts for selected word pairs from the word similarity benchmark SimLex-999 (Hill et al., 2015) and measures the graded contextual effect. We use the English dataset from this task. Its first subtask, **CoSimlex-I**, evaluates the change in similarity between the same word pair under different contexts. As it requires a system to capture different contextual representations of the same word in order to correctly predict the change of similarity to the other word in the pair, CoSimlex-I indirectly measures within-word contextual effect and therefore provides our third *Within Word* task. The second CoSimLex subtask, **CoSimlex-II**, is an *Inter Word* task as it requires a system to predict the absolute gold rating of each word pair consisting of different words in each context. We also evaluate on another related *Inter Word* task, Stanford Contextual Word Similarity (**SCWS**), which provides graded similarity scores of word pairs in independent contexts. Compared with the two *Inter Word* tasks, the three *Within Word* tasks are more sensitive to contextual effects since they penalize strongly a static model (eg. FastText) as being no better than a random baseline. By contrast, we might expect a context-independent static model to perform reasonably, though not as good as a context-sensitive model, in $InterWord$ tasks (Armendariz et al., 2019).

Results: Table 1 reports the performance of each contextualized model before and after the transformation guided by each of the other contextual/static embeddings. In this table, → indicates the direction of the transformation. For example, RoBERTa → FastText denotes using FastText as the target space to transform RoBERTa.

We find that applying transformation is generally able to improve each contextualized model, obtaining the best performance across all the tasks. In particular, we observe substantial improvements in Usim (ca. 0.04 increase of ρ) and SCWS (ca. 0.03 increase of ρ). The most consistent improvement comes from leveraging static embeddings. This is especially evident in *Inter Word* tasks where transforming towards FastText achieves the best performance but leveraging another contextualized model often brings harm. This suggests that the static embeddings are able to inject better inter-word relations (Wang et al., 2019) into a contextualized model. At the same time, static embeddings consistently improve performance in *Within Word* tasks in 24 out of the total 27 configurations, reassuring us that the contextualization power of the original contextual space is not only preserved but even enhanced. Overall, FastText is the most robust target space as it improves all the contextualized source representations for all the tasks except for XLNet in WiC. SGNS and GloVe are also competitive especially in improving *Within Word* tasks.

Analysis: The overall improvement in both *Within Word* and *Inter Word* tasks suggests two possible benefits from the transformation: better within-word contextualization and better overall inter-word semantic space. We perform controlled studies that test for these two sources of improvement in isolation. We test on the best base contextualized space (RoBERTa) with the various transformations.

The fact that a static embedding (FastText) performs better than a random baseline in *Within Word* tasks (see Table 1) suggests that there are some lexical cues in the target words (eg. morphological variations) that can help solve the task alongside the context. To highlight the improvement in contextualization alone, since the *Within Word* tasks before lemmatization may contain different word forms of the same lemma as the target words in each pair, we lemmatize all the target words in the dataset. As a result, each pair in the *Within Word* tasks now contains the identical target word. We observe that the results after lemmatization are slightly lower than before but the transformation especially towards static embeddings is indeed able to improve the contextualization across all the tasks (Table 2).

To test solely the effect on the overall inter word semantic space of the contextualized model becomes better after the transformation, we 'decontextualize' the model by evaluating only on the static anchors of the contextualized embeddings. These static anchors are not sensitive to a particular context and can thus only reflect overall inter word semantic space like embeddings from a static model. We observe improvement from the transformation on the static anchors in *Inter Word* tasks. In particular, aligning towards FastText brings the largest and the most consistent gains. This suggests that FastText may have offered a better ensemble space with RoBERTa and results in a better overall inter word semantic space.

	Within Word			Inter Word	
	Usim (ρ)	WiC (acc%)	CoSimlex-I (r)	CoSimlex-II (ρ)	SCWS (ρ)
Random	0.	0.	50.	0.	0.
FastText	0.1290	56.21	0.2776	0.4481	0.6782
RoBERTa	0.6196	68.28	0.7713	0.7249	0.6884
→ BERT	0.6529	68.21	**0.7814**	0.7087	0.6938
→ XLNet	0.6371	67.50	0.7622	0.6977	0.6689
→ FastText	0.6544	69.00	0.7794	**0.7344**	**0.7159**
→ SGNS	0.6473	**70.07**	0.7761	0.7140	0.7009
→ GloVe	**0.6556**	67.85	0.7783	0.7254	0.6763
BERT	0.5995	66.29	0.7595	0.7228	0.7305
→ RoBERTa	0.6185	66.71	0.7684	0.7172	0.7276
→ XLNet	0.6165	66.57	0.7633	0.7103	0.7196
→ FastText	0.6388	67.57	0.7701	**0.7315**	**0.7507**
→ SGNS	0.6371	**68.28**	**0.7712**	0.7224	0.7421
→ GloVe	**0.6403**	66.79	0.7710	0.7311	0.7327
XLNet	0.4944	63.14	0.7727	0.7450	0.7047
→ BERT	**0.5382**	62.35	**0.7842**	0.7414	0.7369
→ RoBERTa	0.5185	62.64	0.7791	0.7430	0.7230
→ FastText	0.5223	62.50	0.7805	**0.7473**	**0.7563**
→ SGNS	0.5313	**63.71**	0.7780	0.7338	0.7481
→ GloVe	0.5349	62.14	0.7824	0.7411	0.7246

Table 1: Performance on context-aware lexical semantic tasks before and after adjusting RoBERTa, BERT and XLNet to other static (red rows) and contextualized embeddings (blue rows). A static embedding baseline (FastText) is also provided. BERT and RoBERTa are reported as the best models without external resources in WiC (Pilehvar and Camacho-Collados, 2019) and Usim (Garí Soler et al., 2019); the previous best reported score is 0.693 (Neelakantan et al., 2014) for SCWS. (r: uncentered Pearson correlation, ρ: Spearman correlation, acc: Accuracy)

	Within Word			Inter Word	
	Usim (ρ)	WiC (acc%)	CoSimlex-I (r)	CoSimlex-II (ρ)	SCWS (ρ)
RoBERTa (lemma)	0.5657	66.35	0.7305	0.6884	0.6693
→ BERT	0.6189	68.07	0.6884	0.6850	0.6727
→ XLNet	0.6022	66.93	0.7358	0.6716	0.6501
→ FastText	0.6260	68.36	**0.7666**	**0.7150**	**0.7000**
→ SGNS	0.6169	**68.85**	0.7636	0.6960	0.6863
→ GloVe	**0.6277**	68.42	0.7535	0.6925	0.6558
RoBERTa (lemma decon)	-	-	-	0.4894	0.5994
→ BERT	-	-	-	0.4945	0.6310
→ XLNet	-	-	-	0.4868	0.5940
→ FastText	-	-	-	**0.5073**	0.6497
→ SGNS	-	-	-	0.4847	**0.6518**
→ GloVe	-	-	-	0.5016	0.6378

Table 2: Controlled experiments on lemmatised (lemma) and decontextualized (decon) RoBERTa before and after transformation towards static embeddings (red rows) or another contextualized embedding (blue rows). The lemma decon condition in the *Within Word* task is irrelevant as the results will be equivalent to random baselines. (r: uncentered Pearson correlation, ρ: Spearman correlation, acc: Accuracy)

To better understand the contextualization improvement in the *Within Word* scenario, we focus on WiC to perform more detailed analysis. While we report results on the test set in Table 1, we present the following analysis on the train and development sets because the test set labels have not been released. We focus on the best performing model in the task, RoBERTa → SGNS, to examine the difference before and after the transformation.

Overall, we observe a trend for the within-word contextual representations to move slightly closer to each other after the transformation, as the mean cosine similarity of a pair's contextual word representations across all instances has increased from 0.516 to 0.542. We further break down cases according to their labels and find that the transformation mainly brings the representations closer for TRUE pairs (where the context pairs of the word are indeed expressing the same meaning) with the mean cosine similarity increased from 0.606 to 0.651. For FALSE cases where the context pairs refer to distinct meanings of the target word, there is less increase in similarity (from 0.426 to 0.433). We also find that the improved performance in this task can be largely attributed to correcting many erroneous FALSE predictions in the original space as these representations are drawn closer after the transformation (See Appendix D). We qualitatively examine these corrected TRUE cases (Examples are provided in Appendix E), and found that the improvement typically comes from reduced variance for the contextual representations of monosemous words. An example is the word *daughter*. We observe very low cosine similarity among its contextual representations in the original space. These representations are drawn closer after the transformation (eg. cosine similarity from 0.48 to 0.67). We suspect this might be related to contextualized models' over-sensitivity to context changes (Shi et al., 2019).

To summarize the analysis, our controlled experiments confirm our two hypotheses that the transformation brings two independent effects: improved overall inter-word semantic space and improved within-word contextualization. Our qualitative analysis shows that the improved within-word contextualization is likely to be the result of context variance reduction.

5 Related Work

It has been shown that combining different static word representations (for example through averaging or concatenation) into a meta embedding can usually lead to better lexical representations (Coates and Bollegala, 2018; Yin and Schütze, 2016). While these task-independent meta embedding techniques are mainly applied on static embeddings, research has started to explore leveraging ensemble contextualized models when performing fine-tuning on a specific task (Devlin et al., 2019; Xu et al., 2020). Our method, as a post-processing transformation over task-independent contextual representations, is inherently different from these meta embedding and ensemble approaches. Computationally, our method does not require maintaining multiple models at test time, and is therefore more efficient. Our method is also by far the most effective way to leverage static embeddings to improve contextualized representations.[6]

Our methodology is related to studies on aligning cross-lingual embeddings (Doval et al., 2018; Liu et al., 2019a; Schuster et al., 2019). While these works mainly focus on obtaining better cross-lingual representations, our study is the first attempt to show that some of the cross-lingual alignment methods can be applied to improve monolingual contextualized representations with no manual resources required.

6 Conclusion

We present an effective post-processing method that transforms and enhances contextual word representations through static anchors with guidance from other contextualized/static embeddings. We show leveraging static embeddings, with no labeled data, consistently improves (across almost all configurations) on both *Inter Word* and surprisingly *Within Word* context-aware lexical semantic tasks. We also perform controlled analysis to highlight, in isolation, the improvement from the transformation on both contextualization and on an overall inter-word semantic space. In the future, we plan to apply the transformed representations on more lexical semantics tasks such as word sense disambiguation within an application (Navigli and Vannella, 2013).

[6] A simple meta embedding baseline that concatenates contextualized and static representations generally impairs the performance. (Appendix F)

Acknowledgments

We thank the anonymous reviewers for their helpful feedback on this work. We acknowledge Peterhouse College at University of Cambridge for funding Qianchu Liu's PhD research. The work was also supported by the ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (no 648909) awarded to Anna Korhonen. We also appreciate many helpful discussions and feedback from our colleagues in the Language Technology Lab.

References

Carlos Santos Armendariz, Matthew Purver, Matej Ulčar, Senja Pollak, Nikola Ljubešić, Mark Granroth-Wilding, and Kristiina Vaik. 2019. Cosimlex: A resource for evaluating graded word similarity in context. *arXiv preprint arXiv:1912.05320*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Joshua Coates and Danushka Bollegala. 2018. Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 194–198, New Orleans, Louisiana. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, and Steven Schockaert. 2018. Improving cross-lingual word embeddings by meeting in the middle. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 294–304, Brussels, Belgium. Association for Computational Linguistics.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2013. Measuring word meaning in context. *Computational Linguistics*, 39(3):511–554.

Aina Garí Soler, Marianna Apidianaki, and Alexandre Allauzen. 2019. Word usage similarity estimation with sentence representations and automatic substitutes. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 9–21, Minneapolis, Minnesota. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Qianchu Liu, Diana McCarthy, Ivan Vulić, and Anna Korhonen. 2019a. Investigating cross-lingual alignment methods for contextualized embeddings with token-level evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 33–43, Hong Kong, China. Association for Computational Linguistics.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Roberto Navigli and Daniele Vannella. 2013. SemEval-2013 task 11: Word sense induction and disambiguation within an end-user application. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 193–201, Atlanta, Georgia, USA. Association for Computational Linguistics.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. 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 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Weijia Shi, Muhao Chen, Pei Zhou, and Kai-Wei Chang. 2019. Retrofitting contextualized word embeddings with paraphrases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1198–1203, Hong Kong, China. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008.

Yile Wang, Leyang Cui, and Yue Zhang. 2019. How can bert help lexical semantics tasks?

Yige Xu, Xipeng Qiu, Ligao Zhou, and Xuanjing Huang. 2020. Improving bert fine-tuning via self-ensemble and self-distillation. *arXiv preprint arXiv:2002.10345*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1351–1360, Berlin, Germany. Association for Computational Linguistics.

A Details on extracting contextual word representations

We take the average of the last 12 layers from BERT and RoBERTa, and of the full 24 layers from XLNet as feature representations. We empirically found averaging the last 12 layers performs better than averaging the full 24 layers or the last layer for BERT and RoBERTa. This is in line with Tenney et al. (2019) that found semantic information is better captured in higher layers in BERT.

If a word is split into subwords after tokenization, we average the subword representations. We leave other ways of extracting the features for future work.

B Implementation details

For the input pre-trained models, we report their hyper-parameters and training details in the following:

Models	**Hyper-parameters & Training details**
RoBERTa	1024 dimensions; 24 layers; 16-heads, 355M parameters.
BERT	1024 dimensions; 24 layers; 16-heads, 340M parameters.
XLNet	1024 dimensions; 24 layers; 16-heads, 340M parameters.
Static models (eg. FastText)	300-d vectors trained on the latest English Wikipedia. We pad these vectors to 1024 to match the dimension size of the contextualized models.

As to our transformation method, we report the following details:

no. of parameter	1024*1024
Average runtime	10 seconds
Computing infrastructure	GeForce GTX 1080 Ti

We release our code at https://github.com/qianchu/adjust_cwe.git

C Details for experiments and data sets in this study

We provide the statistics for each task including number of examples, train/dev/test splits, and links to downloadable versions of the data in Table 3.

We also report the validation performance for WiC as the only supervised task in our study. Results on the development set and the hyper-parameters (the cosine similarity threshold) are listed in Table 4. The threshold is searched with 0.01 step size until we find the model that achieves the highest accuracy.

D Changes in model prediction on WiC before and after the transformation

Table 5 lists the changes in predicted labels after transforming RoBERTa towards SGNS. We categorize the changes into four groups according to the prediction changes after the transformation. The largest group contains 153 cases that were originally predicted as false negatives and were corrected after the transformation.

E Examples of corrected TRUE cases after transformation in WiC

Below are examples that were corrected to TRUE labels after the transformation in RoBERTa → SGNS. We also report changes in the cosine similarity of contextual representations in each example.

Examples	**similarity change**
I [know] it 's time. It is vital that he not [know].	0.42→0.65
I already have a son , so I would like to have a [daughter]. Her [daughter] cared for her in her old age.	0.48→0.67

F Results on concatenating contextualized and static embeddings

Please refer to Table 6 for a simple baseline that concatenates FastText and each of the contextualized model. We report results for FastText only as it has proved to be the most robust static embedding target space. We found similar results for concatenating with other static embeddings. In short, the simple concatenation with a static embedding generally brings more harm for the contextualized model.

Task	Statistics	Links
Usim	We exclude one sentence that caused xml parsing errors (call.v.1211) and 9 pairs involving this sentence. The final testset contains 1133 context pairs.	http://www.dianamccarthy.co.uk/downloads/WordMeaningAnno2012/
WiC	We use the original dataset with train/development/test splits containing 5428/638/1400 instances respectively.	Link to train and development sets: https://pilehvar.github.io/wic/ Link to test set: https://competitions.codalab.org/competitions/20010
CoSimlex	Contains 333 word pairs and each pair has two different contexts. We test on the whole dataset	https://competitions.codalab.org/competitions/20905
SCWS	2003 word pairs with contexts	shorturl.at/swMS3

Table 3: Statistics for each task evaluated in the study

	WiC (acc%)	threshold
RoBERTa	67.24	0.5300
→ BERT	67.86	0.5600
→ XLNet	67.40	0.5500
→ FastText	68.49	0.5700
→ SGNS	69.12	0.5600
→ GloVe	69.59	0.6000
BERT	68.65	0.5500
→ RoBERTa	68.34	0.5400
→ XLNet	68.80	0.5400
→ FastText	68.65	0.5400
→ SGNS	69.44	0.5500
→ GloVe	68.18	0.5600
XLNet	62.70	0.5300
→ BERT	62.70	0.5900
→ RoBERTa	62.54	0.5800
→ FastText	62.07	0.5700
→ SGNS	61.91	0.5900
→ GloVe	61.75	0.6600

Table 4: Performance on the development set of WiC

	TRUE (gold)	FALSE (gold)
FALSE (before) → TRUE (after)	153	99
TRUE (before) → FALSE (after)	116	145

Table 5: Change of predicted labels after the transformation of RoBERTa → SGNS. The left most column shows the predicted labels before and after the transformation, and the top row shows the gold label. The shaded cells report the number of cases corrected by the transformation.

	Within Word			Inter Word	
	Usim ρ	WiC acc%	CoSimlex-I r	CoSimlex-II ρ	SCWS ρ
RoBERTa	0.6196	68.28	0.7713	0.7249	0.6996
→ FastText	**0.6544**	**69.00**	**0.7794**	**0.7344**	0.7159
+ FastText	0.3301	66.85	0.5854	0.587	**0.7179**
BERT	0.5995	66.29	0.7595	0.7228	0.7520
→ FastText	**0.6388**	**67.57**	**0.7701**	**0.7315**	**0.7507**
+ FastText	0.3663	65.64	0.764	0.6763	0.7488
XLNet	0.4944	**63.14**	0.7727	0.7450	0.7242
→ FastText	**0.5223**	62.50	**0.7805**	**0.7473**	**0.7563**
+ FastText	0.2792	61.21	0.7641	0.6688	.7363

Table 6: Comparing concatenation (+) and our transformation method (→) on leveraging static embeddings. While concatenation may sometimes achieve slightly better results in SCWS, it largely worsens the performance in general. By contrast, our method achieves the most consistent and robust improvements (r: uncentered Pearson correlation; ρ: Spearman correlation; acc: Accuracy)