Enhancing Out-of-Vocabulary Estimation with Subword Attention

Raj Patel and Carlotta Domeniconi

Department of Computer Science
George Mason University
4400 University Dr, Fairfax, VA 22030
{rpatel17, cdomenic}@gmu.edu

Abstract

Word embedding methods like word2vec and GloVe have been shown to learn strong representations of words. However, these methods only learn representations for words in the training corpus and therefore struggle to handle unknown and new words, known as out-of-vocabulary (OOV) words. As a result, there have been multiple attempts to learn OOV word representations in a similar fashion to how humans learn new words, using word roots/subwords and/or surrounding words. However, while most of these approaches use advanced architectures like attention on the context of the OOV word, they tend to use simple structures like ngram addition or character based convolutional neural networks (CNN) to handle processing subword information. In response to this, we propose *SubAtt*, a transformer based OOV estimation model that uses attention mechanisms on both the context and the subwords. In addition to attention, we also show that pretraining subword representations also leads to improvement in OOV estimation. We show *SubAtt* outperforms current state-of-the-art OOV estimation models.

1 Introduction

Word embeddings are very useful in natural language processing tasks. Methods like word2vec (Mikolov et al., 2013a,b) and GloVe (Pennington et al., 2014) train strong semantic representations of words using co-occurrence statistics on a large text corpus, and have been shown to be effective at semantically representing text data. However, one weakness of these methods is that they only learn representations for words that exist in the training corpus, and therefore have no representations on unknown terms, known as out-of-vocabulary (OOV) words. Contextualized embeddings like BERT (Devlin et al., 2018) also suffer from weak performance on rare and unknown words, despite being able to build a contextualized representation of them (Schick and Schütze, 2020). Therefore learning representations for OOV words is an important endeavour. In this work, we focus on static embeddings, where a large amount of OOV work is focused on, and leave contextualized embedding to future work.

Current approaches combine subword and context information to estimate OOV words. While these approaches apply attention mechanisms to aggregate context representations, they tend to do very little with subword representations. As a result, this paper proposes *SubAtt*, a deep neural network attention model that estimates OOV word representations using attention layers (Vaswani et al., 2017) on the subwords in addition to the contexts. *SubAtt* also pretrains subword representations, allowing it to learn quality representations before combining it with context. We show that both pretraining and applying attention on subwords improves OOV estimates, and show that *SubAtt* generally outperforms state-of-the-art OOV estimation models in both intrinsic and extrinsic tasks.

2 Related Work

There are multiple strategies to estimate OOV embeddings. Some OOV strategies use word roots of the OOV word to estimate OOV embeddings (Bojanowski et al., 2017; Pinter et al., 2017; Sasaki et al., 2019) while other methods use the OOV word's context (Lazaridou et al., 2017; Horn, 2017; Herbelot and Baroni, 2017; Arora et al., 2017; Mu and Viswanath, 2018; Khodak et al., 2018). However, more recent attempts combine both subwords and context approaches. Schick and Schütze propose the Form-Context model (Schick and Schütze, 2019c), which estimates OOV embeddings by combining the sum of ngram embeddings (learned by the model) with the sum of word embeddings in the contexts multiplied by a weight matrix (also learned by the model). This model has been extended to the Attentive Mimicking model (Schick

Findings of the Association for Computational Linguistics: ACL 2023, pages 3592–3601
July 9-14, 2023

and Schütze, 2019a) which adds an attention mechanism to the context calculations. A second combined approach is the attention based hierarchical context encoder, known as HiCE (Hu et al., 2019). HiCE is a transformer based model that leverages the hierarchical structure of contexts, using a transformer encoder to encode each context sentence into a sentence embedding, and then using another transformer encoder to combine each sentence embedding into a full context embedding. It estimates subword information using a character based convolutional neural network (CNN), and then combines each piece into a final OOV embedding. HiCE also adapts its model to the OOV word's corpus using Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017). Another approach, Estimator Vectors (Patel and Domeniconi, 2020), trains its own word embeddings, along with subword and context embeddings for OOV estimation. BERTRAM (Schick and Schütze, 2019b) applies an approach similar to the above models, but for contextualized embedding models like BERT (Devlin et al., 2018). While these approaches create strong estimates for OOV words, they have some weaknesses. First, although some use attention mechanisms with the context of an OOV word, none of the aforementioned combined approaches use attention for processing the OOV's subwords[1]. Secondly, none of the static embedding approaches pretrain their subwords; they learn these representations at the same time as the whole model[2]. Therefore, we propose *SubAtt*, a model that uses attention and pretraining on subwords, leading to stronger OOV estimates.

3 SubAtt

We now present *SubAtt*, a transformer (Vaswani et al., 2017) based model for OOV estimation. First, we describe pretraining the subword representations in Section 3.1, then how the model encodes each context sentence in Section 3.2, and finally how *SubAtt* combines subword and context information in Section 3.3.

3.1 Pretraining Subword Representations

First, *SubAtt* learns subword representations for the current set of word embeddings. *SubAtt* learns embeddings for character ngrams of each vocabulary word. This is accomplished by adding a beginning and end special token to the word, and then taking each character subset of that word. We learn representations using the following formulation:

$$sub_{w_t} = \frac{1}{|G_{w_t}|} \sum_{g \in G_{w_t}} z_g \tag{1}$$

where G_{w_t} is the set of character n-grams (the subwords) of the word w_t, and z is the embedding of the subwords. Subword representations z are learned by maximizing the cosine similarity between sub_{w_t} and the corresponding word embedding v_{w_t}. Once these subword representations are trained, they are used in the main *SubAtt* model. An OOV word is broken down into its character ngrams, which are then converted to the set of corresponding subword embeddings Z.

3.2 Context Encoder

SubAtt encodes sentences using a context encoder similar to the one in HiCE (Hu et al., 2019). For each word, an input embedding is built by combining its word embedding and a position embedding. The set of input embeddings for context j (denoted context words Q_j) are then inputted into a transformer encoder:

$$Q'_j = \text{Encoder}(Q_j) \tag{2}$$

which is then averaged for a final context representation c_j. These representations make up the set of context embeddings C.

3.3 Full *SubAtt* Model

SubAtt is composed of a subword half and a context half. The subword inputs Z are the OOV word's ngram subword representations learned in Section 3.1. For the list of contexts, the context representations C are calculated using the architecture described in Section 3.2.

Each type is processed through their own sets of multi-head self attention encoders[3]. (Vaswani

[1] (Sasaki et al., 2019) does use attention on subwords, but the resulting method is not a combined approach, and therefore doesn't show its strength in state-of-the-art settings.

[2] BERTRAM is the exception to this, as it pretrains subwords separately. However, *SubAtt* is the first to do this on static embeddings.

[3] We use different encoders for two reasons; first to avoid input length impacting one input type's influence over another, and secondly to allow different attention calculations to occur for different types; subword tokens may have a different relationship with each other compared to context tokens.

	Bio-NER	Rare-NER	POS	AnEM	MovieMIT	CoNLL 2003
AM	0.6593	**0.1042**	0.3041	**0.2704**	0.5135	0.5848
HiCE	**0.6758**	0.0954	0.4453	**0.2812**	0.5240	0.6105
HiCE 8 Layer	0.6683	0.0972	0.4338	**0.2660**	0.5197	0.6141
SubAtt	**0.6785**	0.0982	**0.5098**	0.2494	**0.5603**	**0.6306**

Table 1: Downstream Tasks - Macro F1 of OOV words. *SubAtt* outperforms or ties other models in most cases.

et al., 2017):

$$Z_{self} = Encoder(Z) \quad (3)$$

$$C_{self} = Encoder(C) \quad (4)$$

Finally, we combined the representations for a final estimate of the OOV embedding. Subwords and contexts can vary in how informative they are to the OOV word, and so it is important to combine them in a fashion that weighs each estimate accordingly. *SubAtt* uses an adaptive weighting strategy used in the Form Context Model and Attentive Mimicking Model (Schick and Schütze, 2019c,a), known as the *gated model*. The subword outputs Z_{self} and context outputs C_{self} are separately averaged into $\widetilde{v}_{subword}$ and $\widetilde{v}_{context}$ respectively. They are then combined by a weighted sum:

$$\widetilde{v}_{final} = \alpha\widetilde{v}_{subword} + (1-\alpha)\widetilde{v}_{context} \quad (5)$$

The weight α is calculated as follows:

$$\alpha = \sigma(w^T[\widetilde{v}_{subword}, \widetilde{v}_{context}] + b) \quad (6)$$

where w and b are learned parameters, and σ is the sigmoid function. $\widetilde{v}_{final}$ is the final estimate of the OOV word embedding. *SubAtt* has eight layers of self attention for the subword inputs and eight layers for the context input.

4 Experiments

4.1 Training Corpus and Word Embeddings

The goal of *SubAtt* is to estimate representations for OOV words given existing word embeddings. For the gold standard word embeddings, we use the embeddings provided by Herbelot and Baroni (Herbelot and Baroni, 2017), as done in previous OOV models like (Schick and Schütze, 2019c) and (Hu et al., 2019). For training models, contexts are taken from the Westbury Wikipedia Corpus (WWC) (Shaoul, 2010). We use the version from (Khodak et al., 2018) with certain words filtered out for the Contextualized Rare Word Task (see Section 4.3). Additionally, as Van Hautte et al. (2019) note, current OOV evaluation tasks benefit from words of the same stem in the training set, even if the original word is filtered out. To combat this, we filter out all words that share a stem with words in the Contextualized Rare Words task similar to the approach in (Van Hautte et al., 2019). The filtered WWC was preprocessed using the preprocessing script provided by Schick and Schütze (2019c), creating a set of words to learn along with context sentences those words appear in. All models are trained using subword and context input from this dataset, by comparing the model's predicted embedding with its gold standard embedding.

4.2 Baselines and Hyperparameters

We now demonstrate the effectiveness of *SubAtt*.[4] We compare it to Attentive Mimicking[5] (AM) and HiCE[6], as they are OOV models that use both subwords and context on existing static word embeddings. Two versions of HiCE are examined; the default with a 2 layer context aggregator, and a version with 8 layers to be more comparable to *SubAtt*. Also, we do not use MAML in the HiCE experiments, in order to focus on how the architecture adapts to multiple OOV tasks. The dataset and vocab are split into a training and validation set for hyperparameter tuning (discussed in more detail in Appendix A).

Ten final trials of each model are trained and then each model is evaluated on various OOV tasks. The results are tested for statistical significance using a one-way ANOVA with a post-hoc Tukey HSD test with a p-value threshold equal to 0.05. The best score is presented in bold, along with any scores that are not significantly different from the best.

4.3 Tasks

We now evaluate *SubAtt* on various OOV tasks. We focus on OOV tasks in English, matching previous work. As *SubAtt* mixes both subwords and

[4]https://github.com/rajicon/SubAtt
[5]https://github.com/timoschick/form-context-model
[6]https://github.com/acbull/HiCE

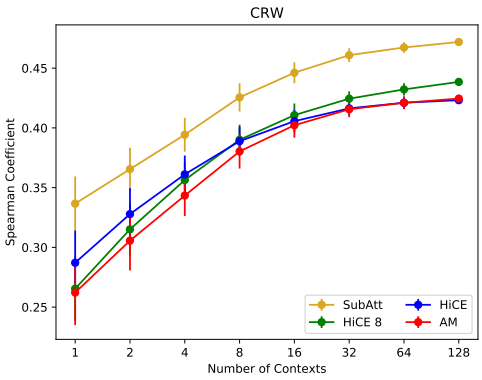

Figure 1: CRW Task

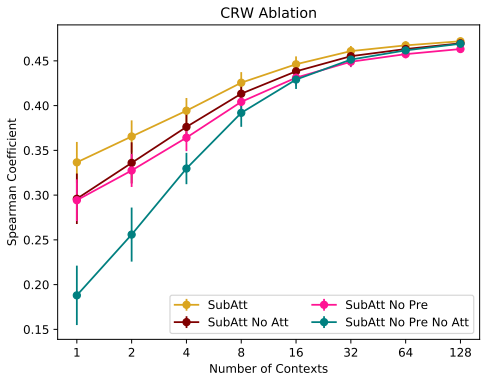

Figure 2: CRW Task - *SubAtt* Ablation

contexts, we select OOV tasks that build OOV representations using both information types.

Contextualized Rare Word Task For intrinsic analysis, we apply the Contextualized Rare Word task (CRW) (Khodak et al., 2018). CRW is built off the Rare Word dataset (Luong et al., 2013), which is a list of rare words paired with other words, along with human similarity scores. Khodak et al. (2018) added contexts to this set, allowing for OOV words to be estimated using both subwords and context. The goal is to output an OOV embedding, compare it to the other words, and evaluate the scores' correlation with human judgements. CRW has a large range of context sizes, from 1 to 128, so the quality and informativeness of the context can vary wildly. However, the words gathered for the Rare Word set have intentionally informative word roots, and therefore we expect subwords to be fairly informative.

The results of the CRW task are shown in Figure 1. *SubAtt* significantly outperforms all competitors in all contexts, showing its effectiveness as an OOV estimator. This shows the strength of pretrained subwords and subword attention.

Downstream Tasks We now demonstrate the strong performance of *SubAtt* embeddings extrinsically, using downstream tasks. In order to focus on OOV words specifically, we choose downstream tasks that output word level labels; specifically named entity recognition and parts-of-speech tagging. For each of these tasks, we train a Bi-LSTM-CRF (Lample et al., 2016), an approach similar to the one in (Hu et al., 2019). The input to these models are normal word embeddings for words in our vocabulary (ones used in training and validation of the original OOV models), and each model's OOV estimates for unknown words.[7] For each dataset, the Bi-LSTM-CRF is trained for 30 epochs 10 times, with the best epoch selected using a validation set each time. This approach is applied to each of the 10 trials of each OOV model. As our focus is estimating OOV words, we report the average test macro F1 score of the OOV words specifically. We also report the results for all words in Appendix B.

We test on 5 named entity recognition tasks: the JNLPBA 2004 Bio-entity recognition dataset (Bio-NER) (Kim et al., 2004), the Rare-NER dataset (Derczynski et al., 2017), the CoNLL 2003 NER dataset (Sang and De Meulder, 2003), AnEM (an anatomy NER dataset) (Ohta et al., 2012), and MovieMIT, a movie querying dataset (Liu et al., 2013). In addition, we test on a parts-of-speech tagging dataset, specifically the Twitter social media POS task (Ritter et al., 2011).

The Downstream Task results are shown in Table 1. *SubAtt* generally outperforms the competitors, strictly winning in 3 of the 6 tasks, and tying for best in one more task, and achieving the second best score in another task. This demonstrates *SubAtt*'s robust and strong performance on OOV words in downstream tasks.

Ablation Analysis We now conduct an ablation study on *SubAtt* in order to demonstrate the impact of the pretraining compared to attention. To this end, we repeat the previous experiments on four variants of *SubAtt*; the original model, the model without attention (*SubAtt* No Att), the model without pretrained subwords (*SubAtt* No Pre), and the model without both (*SubAtt* No Pre No Att). The results are shown in Figure 2 and Table 2.

In the CRW Ablation study, *SubAtt* outperforms the other models. especially in smaller context sizes. This gap closes as the size of contexts in-

[7] OOV words with invalid subwords (no existing character ngrams or no CharCNN characters) are assigned a zero vector.

	Bio-NER	Rare-NER	POS	AnEM	MovieMIT	CoNLL 2003
SubAtt No Pre No Att	0.6503	0.0747	0.3754	0.1904	0.5129	0.5783
SubAtt No Pre	0.6662	**0.0940**	**0.4813**	0.2091	0.5336	0.6081
SubAtt No Att	**0.6721**	0.0898	0.4619	**0.2311**	0.5326	0.6018
SubAtt	**0.6785**	**0.0982**	**0.5098**	**0.2494**	**0.5603**	**0.6306**

Table 2: Downstream Tasks - Macro F1 of OOV words. *SubAtt* outperforms or ties other models in most cases.

creases[8]. This makes sense, as the influence of subwords decreases as our model gains more and more context information, which in turn lowers the impact of the pretraining and attention on subwords in general. Similarly, *SubAtt* performs strongly in the Downstream Ablation, performing the best or tied for the best in all six tasks. The results also demonstrate that pretraining and subword attention individually have a high impact on results, and both combined leads to an even stronger improvement.

5 Conclusion

We propose *SubAtt*, an attention based model that estimates OOV words by using pretrained subword embeddings and subword attention. We show through various experiments that this model estimates more accurate representations of OOV words.

6 Limitations

One limitation of this work is the lack of diversity in the downstream tasks. In our experiments, five of the tasks are named entity recognition and one is a parts-of-speech task. As OOV words make up a small portion of a sentence in the text for our downstream tasks, (the number of OOV words with valid characters ranges from 3% to 12%), analyzing the impact of higher quality OOV estimates is not trivial. For example, in document classification, the predictions depend on each word in the document, and thus the evaluation of OOV estimation will not just be based on the quality of the OOV embeddings, but also on their effect on the result compared to known embeddings. This makes assessing OOV estimation quality more challenging. As such, it is better to focus on tasks with word level output, so the quality of the OOV estimates can be directly judged. However, this limits the types of downstream tasks being analyzed. A second limitation is the fact that all tasks use the English language. As subword impact is dependent on the morphology of a language, the contributions of subword pretraining and attention will vary with the language. However, previous OOV works evaluate on English tasks, and as a result for comparison this paper does the same.

[8]For easier comparison, we report the actual values of the CRW Ablation in Appendix C.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics*, 5(1):135–146.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.

Aurélie Herbelot and Marco Baroni. 2017. High-risk learning: acquiring new word vectors from tiny data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 304–309.

Franziska Horn. 2017. Context encoders as a simple but powerful extension of word2vec. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 10–14.

Ziniu Hu, Ting Chen, Kai-Wei Chang, and Yizhou Sun. 2019. Few-shot representation learning for out-of-vocabulary words. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4102–4112.

Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon M Stewart, and Sanjeev Arora. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Citeseer.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Angeliki Lazaridou, Marco Marelli, and Marco Baroni. 2017. Multimodal word meaning induction from minimal exposure to natural text. *Cognitive science*, 41:677–705.

Jingjing Liu, Panupong Pasupat, Yining Wang, Scott Cyphers, and Jim Glass. 2013. Query understanding enhanced by hierarchical parsing structures. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 72–77. IEEE.

Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective post-processing for word representations. In *6th International Conference on Learning Representations, ICLR 2018*.

Tomoko Ohta, Sampo Pyysalo, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Open-domain anatomical entity mention detection. In *Proceedings of the workshop on detecting structure in scholarly discourse*, pages 27–36.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Raj Patel and Carlotta Domeniconi. 2020. Estimator vectors: Oov word embeddings based on subword and context clue estimates. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

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.

Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. 2017. Mimicking word embeddings using subword RNNs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 102–112.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1524–1534.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Shota Sasaki, Jun Suzuki, and Kentaro Inui. 2019. Subword-based compact reconstruction of word embeddings. 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 3498–3508.

Timo Schick and Hinrich Schütze. 2019a. Attentive mimicking: Better word embeddings by attending to informative contexts. 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 489–494.

Timo Schick and Hinrich Schütze. 2019b. Bertram: Improved word embeddings have big impact on contextualized model performance. *arXiv preprint arXiv:1910.07181*.

Timo Schick and Hinrich Schütze. 2019c. Learning semantic representations for novel words: Leveraging both form and context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6965–6973.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8766–8774.

Cyrus Shaoul. 2010. The Westbury lab Wikipedia corpus. *Edmonton, AB: University of Alberta*, page 131.

Jeroen Van Hautte, Guy Emerson, and Marek Rei. 2019. Bad form: Comparing context-based and form-based few-shot learning in distributional semantic models. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 31–39.

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 *Advances in neural information processing systems*, pages 5998–6008.

A Hyperparameter Selection and Implementation

For the competitors, we use the implementations from the original authors. Attentive Mimicking was implemented in Tensorflow (Abadi et al., 2015), while HiCE, *SubAtt*, and the Downstream Bi-LSTM-CRF were implemented and trained using Pytorch (Paszke et al., 2019). *SubAtt* uses transformer code implemented in the HiCE (Hu et al., 2019) code.

All model (*SubAtt* and competitors) hyperparameters were determined using Grid Search, between learning rates 0.00001, 0.0001, 0.001, 0.01, and subword dropout probabilities from 0.0 to 90.0, incremented every 10.0. The hyperparameters with the best validation loss were chosen.

For training, we use the Adam optimizer and train each model for 30 epochs, saving each epoch and picking the model with the best validation loss. The training size was around 300,000 and the validation size was around 78,000 (there is some variance in how many examples are generated).

B Downstream Task Full Results

The results of the Downstream Tasks for all words (not just the OOV ones) are reported in Table 3. In addition, the Ablation Downstream results are shown in Table 4. These results are a lot closer, which is to be expected, as most words are not OOV and therefore use the standard embeddings. Despite being close, *SubAtt* still performs the best or ties for the best in all tasks. We also emphasize that although OOV improvement does not make a huge difference in the overall results, it is important to note that looking at overall averages may not be the best way to measure impact, as OOV/unknown words tend to be more important to the meaning of a sentence, as domain specific terms or proper nouns are often OOV words. This is especially true in problems like Named Entity Recognition.

C Ablation CRW Values

For clarity, in Table 5 we report the numerical values corresponding to the plots given in Figure 2.

	Bio-NER	Rare-NER	POS	AnEM	MovieMIT	CoNLL 2003
AM	**0.7350**	0.1788	0.6068	**0.4156**	**0.6551**	**0.5356**
HiCE	**0.7373**	**0.1872**	**0.6407**	**0.4227**	0.6469	**0.5380**
HiCE 8 Layer	**0.7369**	**0.1927**	**0.6390**	0.4110	0.6481	**0.5400**
SubAtt	**0.7379**	**0.1890**	**0.6447**	**0.4286**	**0.6595**	**0.5405**

Table 3: Downstream Tasks - Macro F1 scores of All Words in Task

	Bio-NER	Rare-NER	POS	AnEM	MovieMIT	CoNLL 2003
SubAtt No Pre No Att	0.7269	**0.1859**	0.6326	**0.4169**	0.6419	**0.5365**
SubAtt No Pre	**0.7323**	**0.1913**	0.6325	**0.4182**	0.6462	**0.5379**
SubAtt No Att	**0.7368**	**0.1880**	**0.6477**	**0.4263**	**0.6509**	**0.5363**
SubAtt	**0.7379**	**0.1890**	**0.6447**	**0.4286**	**0.6595**	**0.5405**

Table 4: Downstream Tasks - Macro F1 scores of All Words in Task

	1	2	4	8	16	32	64	128
SubAtt No Pre No Att	0.1880	0.2559	0.3297	0.3919	0.4291	0.4514	0.4617	0.4691
SubAtt No Pre	0.2943	0.3275	0.3642	0.4042	0.4312	0.4490	0.4575	0.4632
SubAtt No Att	0.2959	0.3360	0.3762	0.4133	0.4383	0.4552	**0.4632**	**0.4696**
SubAtt	**0.3366**	**0.3655**	**0.3943**	**0.4256**	**0.4462**	**0.4609**	**0.4673**	**0.4719**

Table 5: CRW Ablation

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 6.

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 4

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The data set used was publicly available, and intended for evaluation.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The text data does not contain personal information. It is built out of available, general text.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☒ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☑ Did you run computational experiments?

Left blank.

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
In terms of GPU hours and computing infrastructure, the implementations of competitors varied, and so it would hard to include a "fair" comparison.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4 and Appendix A

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix A.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.