Subword Attention and Post-Processing for Rare and Unknown Contextualized Embeddings

Raj Patel
Department of Computer Science
George Mason University
4400 University Dr, Fairfax, VA 22030
rpatel17@gmu.edu

Carlotta Domeniconi
Department of Computer Science
George Mason University
4400 University Dr, Fairfax, VA 22030
cdomenic@gmu.edu

Abstract

Word representations are an important aspect of Natural Language Processing (NLP). Representations are trained using large corpora, either as independent static embeddings or as part of a deep contextualized model. While word embeddings are useful, they struggle on rare and unknown words. As such, a large body of work has been done on estimating rare and unknown words. However, most of the methods focus on static embeddings, with few models focused on contextualized representations. In this work, we propose SPRUCE, a rare/unknown embedding architecture that focuses on contextualized representations. This architecture uses subword attention and embedding post-processing combined with the contextualized model to produce high quality embeddings. We then demonstrate these techniques lead to improved performance in most intrinsic and downstream tasks.

1 Introduction

Word representations are an important aspect of NLP. While initially, word embeddings were trained separately and inserted into task specific architectures ("static" embeddings), modern approaches use deep architectures to generate contextualized representations (Devlin et al., 2018; Peters et al., 2018; Liu et al., 2019). A weakness of static representations is that they only exist for a trained vocabulary; there are no representations for unknown words. While deep contextualized models can theoretically produce a new representation, Schick and Schütze (2020) demonstrated that these representations for unknown/rare words are of poor quality, implying that rare/unknown words are still a challenge for contextualized embeddings. In response, there have been attempts to create new representations for these words. While there has been a large body of work on static embeddings, less has been focused on contextualized embeddings, especially approaches that incorporate recent innovations enhancing static rare/unknown estimation. Motivated by this, we propose a new architecture for rare/unknown estimation of contextualized embeddings. This model incorporates subword attention and embedding post-processing for higher quality estimates for contextualized models. We call this approach **S**ubword Attention and **P**ostprocessing for **R**are and **U**nknown **C**ontextualized **E**mbeddings (SPRUCE). We demonstrate that this model has superior results in most evaluation scenarios.

2 Related Work

Rare/unknown word representations have been well studied in static word embeddings. Early approaches used context sentences to estimate new word embeddings (Herbelot and Baroni, 2017; Lazaridou et al., 2017; Horn, 2017; Arora et al., 2017; Mu and Viswanath, 2018; Khodak et al., 2018), while other approaches use the rare words' morphemes/subwords to estimate the embedding (Bojanowski et al., 2017; Sasaki et al., 2019; Pinter et al., 2017). The most effective approaches combine context sentences and subwords (Schick and Schütze, 2019c,a; Hu et al., 2019; Patel and Domeniconi, 2020, 2023). The combined model *SubAtt* (Patel and Domeniconi, 2023), for instance, uses transformer self attention (Vaswani et al., 2017) on context like other models, but also uses transformer self attention on the subword representations, leading to strong results. Rare/unknown words have also been studied on contextualized embeddings, with the goal of constructing new representations for use in the initial embedding layer of the contextualized deep model. While less-studied than static embeddings, there have been attempts to effectively estimate rare/unknown contextualized embeddings. The current state-of-the-art approach on contextualized models is BERTRAM (Schick and Schütze, 2019b); BERTRAM constructs the context representations using the BERT architecture. It then combines these representations us-

Findings of the Association for Computational Linguistics: NAACL 2024, pages 1383–1389
June 16-21, 2024 ©2024 Association for Computational Linguistics

ing the attention mechanism from Attentive Mimicking (Schick and Schütze, 2019a, 2020). It uses learned subwords to estimate the rare/unknown embedding, and then inputs this estimate into the BERT model for each context sentence. BERTRAM has been shown to output strong rare/unknown embeddings for use in a BERT architecture. However, contextualized rare/unknown words are understudied, and models don't incorporate recent innovations found in static embedding equivalents. In response to this, we propose SPRUCE, a model that incorporates the strengths of previous static models like *SubAtt* and contextualized models like BERTRAM to create a new architecture that is state-of-the-art in most rare/unknown evaluation tasks.

3 Model

We now present SPRUCE[1]. We focus on estimating rare and unknown embeddings with the BERT (Devlin et al., 2018) model, although this can be adapted to any deep model. We combine aspects of the previous state-of-the-art model BERTRAM (Schick and Schütze, 2019b) with attention on the subword input, similar to the one proposed in static word embeddings model *SubAtt* (Patel and Domeniconi, 2023) but has not been previously used in contextualized models. In addition, we train SPRUCE on post-processed embeddings, with top PCA components removed. A diagram of SPRUCE is shown in Figure 1.

3.1 Pretrained Aspects

Similar to BERTRAM, we start with pretraining a context half and a subword half of the model separately. We use the same architectures pretrained in BERTRAM for SPRUCE. For the context half, the architecture uses BERT to encode each context sentence, which it then applies Attentive Mimicking (Schick and Schütze, 2019a) for a final rare word estimate. For the subwords, the architecture learns character n-gram representations and then combines them for a final rare word estimate. These pretrained architectures are then used to build SPRUCE, as discussed in the following sections.

3.2 SPRUCE Context Architecture

Similar to BERTRAM, we extract BERT representations for each context sentence C_i. We then use these to calculate our new representations using Attentive Mimicking (Schick and Schütze, 2019a, 2020):

$$v_{C_i} = BERT(C_i) \tag{1}$$

$$v_{ctx_1} = \sum_{i=1}^{C} \rho(C_i) v_{C_i} \tag{2}$$

where $\rho(C)$ is calculated using the attention mechanism used in Attentive Mimicking (see (Schick and Schütze, 2019a) for more details). Next, we calculate a second context representation, using a transformer encoder self attention layer, denoted as $Encoder_{ctx}$. We take the mean of this result:

$$v_{C_2} = Encoder_{ctx}(v_C, v_C, v_C) \tag{3}$$

$$v_{ctx_2} = \frac{1}{|v_{C_2}|} \sum_i v_{C_{2i}} \tag{4}$$

This approach yields two context representations, v_{ctx_1} and v_{ctx_2}.

3.3 SPRUCE Subword Architecture

Unlike BERTRAM, which creates a subword estimate and then inserts it into each context sentence, we also incorporate the subword representation at the end of the model. In addition, we apply attention on the subwords. This was proposed in (Patel and Domeniconi, 2023) for static embeddings; ours is the first architecture to do this with contextualized ones. We use two subword representations. First, in an effort to match the context processing of BERT, we apply transformer encoder layers to the pretrained subword embeddings. We use 12 layers to match the BERT architecture. We then take the mean of those representations:

$$v_{S_2} = Encoder_{sub_{12}}(v_S, v_S, v_S) \tag{5}$$

$$v_{sub_1} = \frac{1}{|v_{S_2}|} \sum_i v_{S_{2i}} \tag{6}$$

where V_S is the set of character ngram subwords that make up the target rare/unknown word. Secondly, to match the context half of the architecture, we use another transformer self attention layer, and then take the mean:

$$v_{S_3} = Encoder_{sub1}(v_{S_2}, v_{S_2}, v_{S_2}) \tag{7}$$

$$v_{sub_2} = \frac{1}{|v_{S_3}|} \sum_i v_{S_{3i}} \tag{8}$$

This yields two subword representations, v_{sub_1} and v_{sub_2}.

[1] https://github.com/rajicon/SPRUCE

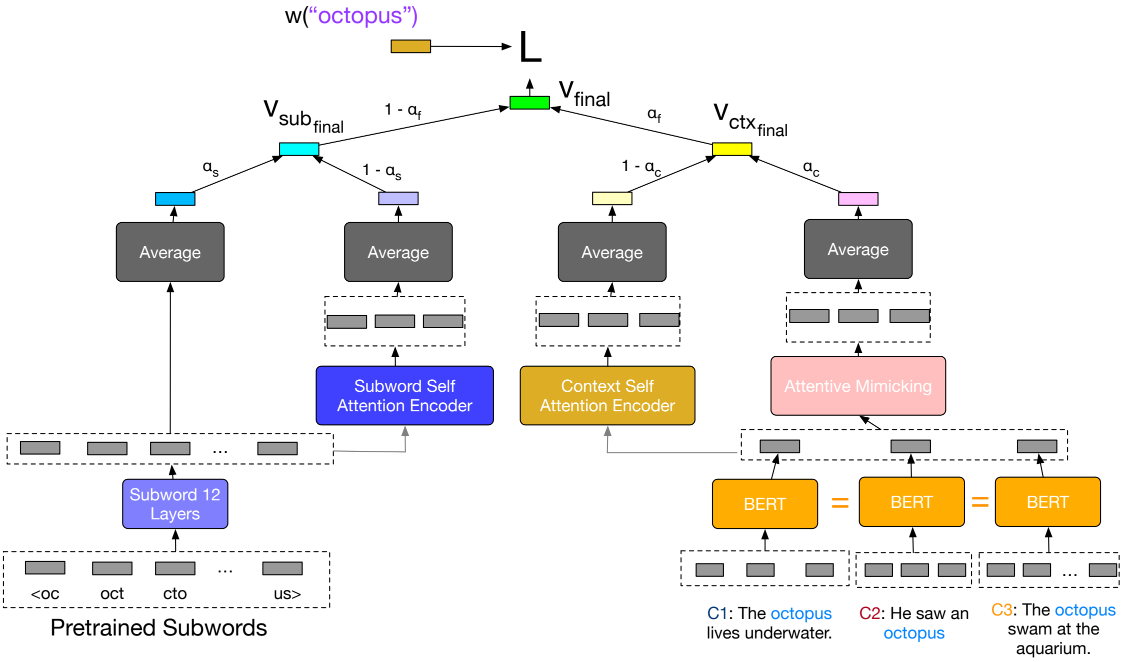

Figure 1: SPRUCE Model Architecture. BERT along with attention blocks are used to create a context representation. Pretrained subwords along with attention blocks are used to create a subword representation. These are then combined using hierarchical gating for a final rare word estimate.

3.4 Combining Subword and Context

We experimented combining the four values in various ways, but found that a hierarchical gating approach worked best. We use gate functions originally proposed in (Schick and Schütze, 2019c), applied multiple times to combine each piece. First, we combine the context representations with each other and the subword representations with each other. We then combine the final context and final subword representations:

$$v_{ctx_{final}} = \alpha_c v_{ctx_1} + (1 - \alpha_c) v_{ctx_2} \quad (9)$$

$$v_{sub_{final}} = \alpha_s v_{sub_1} + (1 - \alpha_s) v_{sub_2} \quad (10)$$

$$v_{final} = \alpha_f v_{ctx_{final}} + (1 - \alpha_f) v_{sub_{final}} \quad (11)$$

with weights of each α is calculated as follows:

$$\alpha_j = \sigma(w_j^T [v_{j1}, v_{j2}] + b) \quad (12)$$

where $w_j \in R^{2d}$ and b is a bias value. Our final representation is v_{final}. During training, this is compared to the original embedding (we refer to this as v_{gold}) using Mean Squared Error as the loss.

3.5 Post-Processing Label Embeddings

Word embeddings tend to share some common directions. These common directions carry little semantic content, and can distract from the meaningful components in embeddings. Mu and Viswanath (2018) and Arora et al. (2017) proposed post-processing word embeddings in order to improve their performance in various tasks. The post-processing approach removes top PCA (Pearson, 1901) components from each embedding, removing less meaningful aspects of the embeddings. While post-processing is generally studied on static word embeddings, Sajjad et al. (2022) demonstrated that this post-processing shows improvement in contextualized embeddings as well. Motivated by this, we propose training SPRUCE on post-processed BERT embeddings. The goal is to train the model to output embeddings that carry meaningful content. Training on post-processed embeddings should force the model to focus on those instead of common directions found in the embeddings. To this end, we remove the top seven components from the BERT embeddings before using them to supervise training. We note that this is only done when training SPRUCE; when inserting the estimated embeddings into the BERT architecture, we do not post-process the common embeddings. The goal is to estimate embeddings that work well in a standard BERT model, and as a result, we do not post-process there.

4 Experiments

4.1 Model Training

We extract gold standard embeddings of frequent words from the embedding layer of the BERT

	Rare	Medium
BERTRAM	0.2852	0.3580
BERTRAM + PCA	0.2902	**0.3721**
SPRUCE	0.2952	0.3483
SPRUCE + PCA	**0.2994**	0.3599

Table 1: WNLaMPro (MRR)

model for use as labels. However, as discussed in (Schick and Schütze, 2020), most embeddings use subword tokenization, and as such, an embedding doesn't exist for all words in the vocabulary. In order to get gold standard embeddings for these words, we use One Token Approximation (Schick and Schütze, 2020). This approach builds an embedding that impacts the BERT model similarly to how the real subword token embeddings do, which is effective for use as a gold standard embedding for the word. We extract context sentences from the Westbury Wikipedia Corpus (WWC) (Shaoul, 2010) for each gold standard word.

4.2 Baselines and Hyperparameters

We compare our approach to BERTRAM[2] (Schick and Schütze, 2019b), the current state-of-the-art. For both models, we pretrain a context only and subword only model, using the same parameters used in (Schick and Schütze, 2019b) with one difference; we increase the subword dropout from 0.1 to 0.3, which we found improved results in both models. We train each model for 10 epochs with a learning rate of 1e-6 (which we found to be best out of 1e-6, 1e-5, and 1e-4). For each model, we train a version based on the standard embeddings, and one trained on post-processed embeddings (denoted "+ PCA"). 10 trials of each model were trained. As we don't have an evaluation set, we test the model saved at each epoch in the evaluation task, and take the best performance. We conduct significance testing using one-way ANOVA with a post-hoc Tukey HSD test. We use a p-value threshold equal to 0.05. We present the best result and any result not significantly different in bold. We also compare each model with its PCA post-processed version, where we present the significant best with an underline.

4.3 Evaluation Tasks

Intrinsic Tasks First, we conduct intrinsic evaluation of our estimated embeddings. The first task we study is the WNLaMPRo task, proposed in (Schick and Schütze, 2020). This task contains various patterns containing vocabulary split by frequency (frequent, medium, and rare). This task then uses simple prompts to measure performance. For example, a frequent pattern may evaluate the word predicted in "A lime is a ", while a similar rare pattern may evaluate the word predicted in "A kumquat is a ". The performance is based on where the real word ranks in the predicted probabilities, measured with Mean Reciprocal Rank (MRR). In our evaluation, we use the models to estimate on rare and medium words, and judge the performance on the new embeddings. We present the results of WNLaMPro in Table 1. As shown in the results, SPRUCE outperforms BERTRAM in rare word performance, but has a weaker performance with medium frequency words. Additionally, we find that PCA post-processing improves both BERTRAM and SPRUCE in both rare and medium words. These results demonstrate SPRUCE's strength at estimating strong rare word representations, along with post-processing label effectiveness at improving embedding performance in both rare and medium words.

Downstream Evaluation While intrinsic evaluation of estimated embeddings is important, the main motivation of using deep contextualized models like BERT is for finetuning on downstream tasks. To this end, we evaluate rare/unknown word performance on various downstream tasks, similar to the procedure done in (Patel and Domeniconi, 2023). However, here we insert the estimated embeddings into a standard BERT model, then finetune the model[3] on the training set (with the best model picked by the validation set). We then evaluate the performance on the test set for that task. Each task presented here is a word level task, which allows us to focus analysis on the rare/unknown words. We focus on six downstream tasks; five NER tasks: AnEM, (Ohta et al., 2012), Bio-NER (Kim et al., 2004), CoNLL 2003 (Sang and De Meulder, 2003), MovieMIT (Liu et al., 2013), and Rare-NER (Derczynski et al., 2017) and one parts-of-speech task POS (Ritter et al., 2011). We present the results on rare and unknown words in Table 2. We find that SPRUCE significantly outperforms BERTRAM in all tasks. This demonstrates SPRUCE's high performance at estimating rare and unknown words. Interestingly, PCA post-processing does not seem to affect results here in

[2] For more model details of BERTRAM and SPRUCE, see Appendix B.

[3] We freeze the embedding layer so we can evaluate the quality of embeddings, not finetuning.

	AnEM	Bio-NER	CoNLL 2003	MovieMIT	POS	Rare-NER
BERTRAM	0.3652	0.7241	0.6617	0.6295	0.2449	0.2592
BERTRAM + PCA	0.3579	0.7252	0.6633	0.6657	0.2346	0.2652
SPRUCE	**0.3867**	**0.7399**	**0.6963**	**0.6801**	**0.4761**	**0.2874**
SPRUCE + PCA	**0.3793**	**0.7409**	**0.6974**	**0.6895**	0.4570	**0.2819**

Table 2: Downstream Tasks - Macro F1 of Rare/Unknown Words

most cases, except for an improvement in BERTRAM in the MovieMIT task and weaker performance in SPRUCE in the POS task. We posit that this lack of impact is due to the fact that post-processing improves estimated embeddings on a finer grained basis. For the downstream tasks, which care more about general features, the improvement gained by post-processing may not have as much impact.

5 Conclusion

We propose SPRUCE, an architecture that uses deep contextualized models to estimate new representations of rare/unknown words for use in those models. We show the strength of SPRUCE in intrinsic and downstream tasks.

Limitations

This work has some limitations. Similar to previous work, task diversity of downstream tasks is limited. Due to ability to focus on rare/unknown words, word level tasks are desirable for analysis, and therefore five out of the six tasks are named entity recognition tasks.

Acknowledgments

This project was supported by resources provided by the Office of Research Computing at George Mason University (URL: https://orc.gmu.edu) and funded in part by grants from the National Science Foundation (Awards Number 1625039 and 2018631).

References

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.

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.

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.

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

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, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

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.

Raj Patel and Carlotta Domeniconi. 2023. Enhancing out-of-vocabulary estimation with subword attention. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3592–3601.

Karl Pearson. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

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.

Hassan Sajjad, Firoj Alam, Fahim Dalvi, and Nadir Durrani. 2022. Effect of post-processing on contextualized word representations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3127–3142.

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.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

A Implementation Details

All experiments were conducted using Pytorch (Paszke et al., 2019) and Huggingface (Wolf et al., 2020) libraries. Our implementation was heavily based on the BERTRAM[4] code.

[4] https://github.com/timoschick/bertram/

B Model Details

Both BERTRAM and SPRUCE are built on top of the BERT base model. The parameters from the BERT architecture along with the learned subword representations make up a large portion of the parameter count. SPRUCE makes use of additional transformer encoder blocks, which increases its parameter count compared to BERTRAM. The final parameter counts for BERTRAM is 176,620,032 and for SPRUCE is 242,803,203.