Aligning Word Embeddings from BERT to Vocabulary-Free Representations

Alejandro Rodriguez Perez
School of Eng. & Computer Science
Dept. of Computer Science
Baylor University
Email: Alejandro_Rodriguez4@Baylor.edu

Korn Sooksatra
School of Eng. & Computer Science
Dept. of Computer Science
Baylor University
Email: Korn_Sooksatra1@Baylor.edu

Pablo Rivas, *Senior, IEEE*
School of Eng. & Computer Science
Dept. of Computer Science
Baylor University
Email: Pablo_Rivas@Baylor.edu

Ernesto Quevedo
School of Eng. & Computer Science
Dept. of Computer Science
Baylor University
Email: Ernesto_Quevedo1@Baylor.edu

Javier Turek
Intel Labs
Intel Corporation
Portland, Oregon
Email: Javier.Turek@Intel.com

Gisela Bichler
School of Criminology & Criminal Justice
California State University
San Bernardino
Email: bichler@csusb.edu

Tomas Cerny
School of Eng. & Computer Science
Dept. of Computer Science
Baylor University
Email: Tomas_Cerny@Baylor.edu

Laurie Giddens
Information Tech. & Decisions Sci. Dept.
G. Brint Ryan College of Business
University of North Texas
Email: Laurie.Giddens@unt.edu

Stacie Petter
School of Business
Management of Information Systems
Wake Forest University
Email: petters@wfu.edu

***Abstract*—This paper investigates the limitations of transformer-based models in handling a fixed vocabulary, which can lead to poor generalization of out-of-vocabulary words and domains. To address this, we explore the use of transfer learning from a vocabulary-rigid transformer to a vocabulary-free one by aligning the word-embedding layer. Our approach trains a CNN to mimic the word embeddings layer of a BERT model, using a sequence of byte tokens as input. By replacing the word embeddings layer of the baseline BERT model with the aligned CNN network, we evaluate the model's generalization performance and ability to handle a broader range of linguistic inputs. Our results demonstrate the advantages of using cosine-based loss functions in the alignment process. Our approach makes important contributions toward developing more flexible and robust NLP models.**

***Index Terms*—model distillation, word embeddings, bert, natural language processing, machine learning, deep learning**

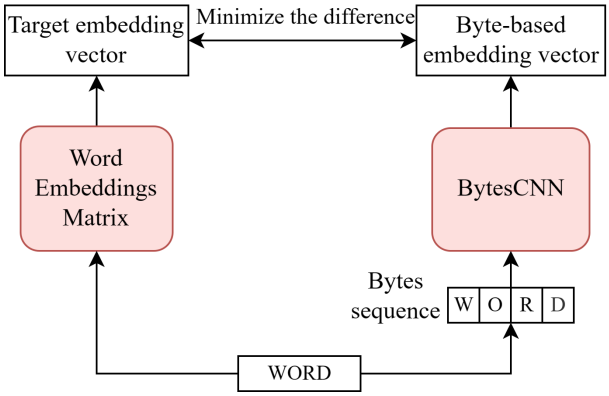

Fig. 1: Embedding alignment framework.

I. Introduction

Transformer-based models have become the de facto standard for many natural language processing (NLP) tasks, thanks to their ability to capture complex linguistic patterns and dependencies [1], [2]. Amongst those, subword-level tokenization is commonplace since it gives the model more flexibility to handle out-of-vocabulary inputs [3], [4].

Recent research has highlighted the shortcomings of subword-level tokenization [5]–[7], one of them being their dependence on a fixed vocabulary, which can lead to poor generalization of out-of-vocabulary words and domains [8], [9]. For instance, forensic NLP models are used to detect covert criminal communications (CCC) hidden within a large volume of text-based interactions. Since CCC typically rely on unusual characters and subwords to obfuscate the meaning of text [10]–[15] rigid vocabulary-based algorithms are limited.

Several solutions have been proposed to address these limitations [16]–[18]. In particular, we highlight those that character-based models to handle a more diverse range of linguistic inputs [19]–[21].

Motivated by the need for more flexible and robust models to handle a wide range of inputs, including out-of-vocabulary words and domain-specific language, we investigate whether we can apply transfer learning from a vocabulary-rigid transformer to a vocabulary-free one by aligning the word-embedding layer, as shown in Figure 1. We use a convolutional neural

network (CNN) that takes as input a sequence of byte tokens and produces a single vector. We then train this network to mimic the word embeddings layer of a Bidirectional Encoder Representations from Transformers (BERT) model [3]. For evaluation, the word embeddings layer of the baseline BERT model is replaced with the aligned CNN network, and evaluation is conducted on the whole network, end-to-end. We hypothesize that this approach improves the model's generalization performance and enables it to handle a broader range of linguistic inputs.

Our approach makes several important contributions toward developing more flexible and robust NLP models. Specifically, this paper: noitemsep

- Investigates transfer learning from a vocabulary-rigid transformer to a vocabulary-free one by aligning the word-embedding layer.
- Advances a method to re-train the embedding layer of transformer-based models by transferring and aligning a byte-based representation with word-based embeddings.
- Assesses different loss functions for the alignment process, showing the advantages of using cosine-based loss functions

This paper is organized as follows: Section II provides an overview of related work, including existing approaches to producing word embeddings, as well as the research problem we address. Section III describes our methodology, including the alignment process for transferring word embeddings to bytes-based embeddings and the training process for the embedding layers. Section IV presents our experiments and results, including a description of the experiments conducted to evaluate our approach, the results of those experiments, and an analysis of the transfer learning capability. Finally, in Section V, we draw conclusions based on our findings and discuss potential future directions for research in this area.

II. Related Work

A. Word Embeddings

In the last decade, word embeddings have become the standard method for word representation in NLP. Those have evolved from simple co-occurrence models like Skip-Gram [22], Continuous Bag of Words(CBOW) [23], and GloVe [24], to contextualized ones like Context2Vec [25] and CoVe [26].

However, research on language models has also sought to develop deep contextual embeddings based on the internal representations of a trained language model. For example, Peters et al. [27] proposed Embedding from Language Models (ELMo). The introduction of the transformer architecture [28] started the explosion of big language models to obtain better and better contextual embeddings like Generative PreTraining [4], Bidirectional Encoder Representations from Transformers (BERT) [3], GPT-2 [4], XLNet [29], RoBERTa [30], ALBERT [31] and BART [32].

The models mentioned before are based on a word or subword-level tokenization, and their embedding matrix needs to account for large vocabulary sizes. Moreover, word and subword-based models are rigidly tied to the vocabulary they were trained on, which renders the task of extending their support to unseen words difficult at the very best. Multiple works proposed to use character-based language models to overcome this limitation [19], [33]–[35]. However, the problem with this approach is that they do not extend to a larger character set easily. As an alternative, researchers have also successfully explored byte-based techniques to represent words [36], [37].

B. Model Distillation

Our work also relates to the concept of knowledge distillation, a method to transfer the knowledge of a complex model into a simpler one [38]. Several methods have been studied. For instance, Sanh et al. [39] showed that it is feasible to distill a pre-trained BERT model's knowledge into a smaller and faster model while retaining 97% of the original model's performance. For a comprehensive survey on knowledge distillation, the reader is referred to [40].

C. Problem Statement

Byte-based transformers have shown great potential to overcome the issues of word and subword-based models. Still, training a transformer model from scratch is an expensive and time-consuming process. Mershad et al. [41] introduced DistillEmb, a method that distills learned word embeddings into a convolutional neural network. The authors use a contrastive learning mechanism based on a triplet loss [42]. The results suggested that DistillEmb represented the words better in morphologically rich languages by interpolating their meanings from their characters. We take inspiration from this work and aim at a similar goal. However, we propose applying this methodology to a transformer model's word embedding layer. A successful application of our methodology allows deriving a byte-based, hence vocabulary-free, transformer model from an existing pre-trained one, thus avoiding the cumbersome pre-training task.

III. Methodology

It has been shown that a large language model (LLM) can be effectively pre-trained using byte-derived word representations [35], [36], [43], which allows moving away from vocabulary-rigid word representations to vocabulary-free ones. But pre-training an LLM from scratch is very resource intensive. Instead, we propose a methodology to transfer learn from a word embedding matrix to a new byte-based embedding network. Our goal is to obtain a vocabulary-free representation while transferring the knowledge existing in a pre-trained LLM to the greatest extent possible.

To achieve that, we define a neural network that processes an arbitrary sequence of bytes and produces a vector of the appropriate embedding size. Then, we train such a byte-based network, using a knowledge distillation setting, to mimic the word embedding matrix's behavior for those words (and subwords) present in the vocabulary of the existing pre-trained LLM, as depicted in Figure 1. We call this process **alignment**.

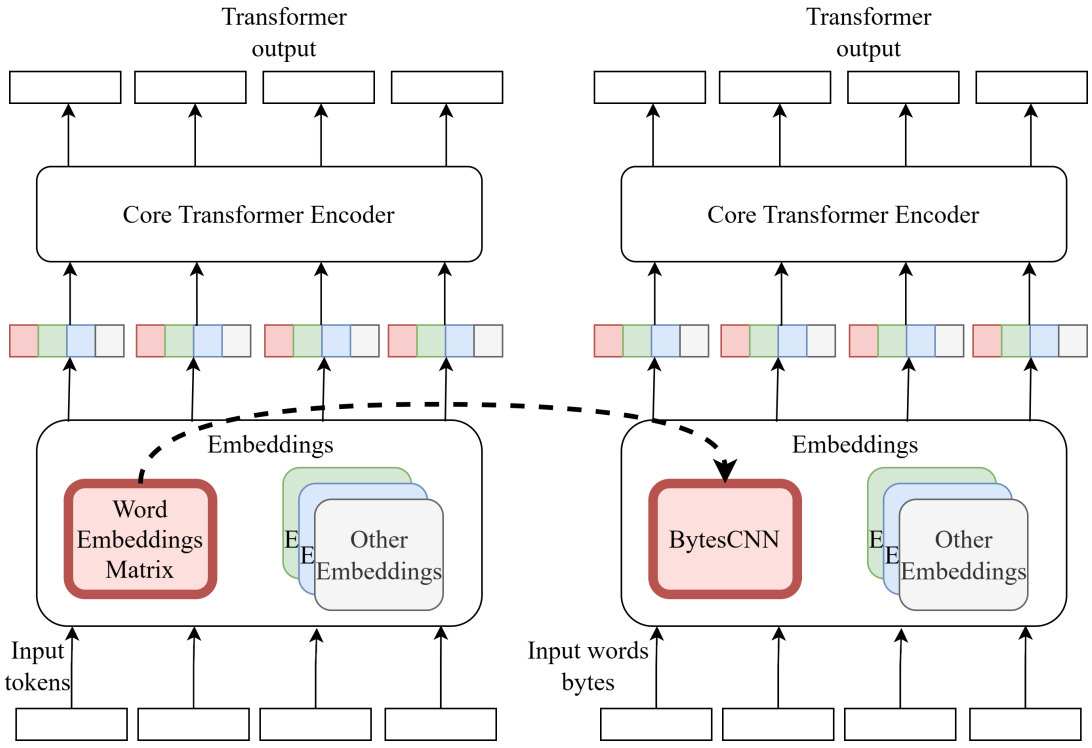

Fig. 2: Illustration of the replacement of the transformer word embedding layer with the aligned BytesCNN.

The byte-based embedding can then be plugged into the LLM in place of the word embedding matrix. This way, the knowledge from a vocabulary-rigid pre-trained LLM is effectively transferred to a vocabulary-free one. Figure 2 illustrates this change to a transformer model.

A. Bytes CNN Architecture

The model we used to substitute the embedding layer was proposed by [34]. We refer to it as BytesCNN instead, because it emphasizes that the input tokens come from a vocabulary of bytes. Thus, the model can process any byte sequence. Figure 3 shows a high-level depiction of the architecture.

Notably, four convolutions with different kernel sizes and numbers of channels are applied in parallel to the input. The results are concatenated after max pooling and ReLU activation. A Highway layer [44] is then applied to the concatenation of the convolutions' outputs. Finally, the vector is projected to the embedding space.

B. Loss Function

We leveraged the use of a variety of loss functions. Apart from standard mean square error (MSE), we tried cosine error and two combinations of MSE and cosine error described later in the subsection.

The embedding vectors in the BERT model are close to laying on the surface of a ball with a radius of approximately 1.41 units and a slight variance (0.19 to be precise) in their norms. The distribution of their norms is illustrated in Figure 4.

This observation supported our intuition to use cosine-based loss functions.

Our first loss function uses a plain cosine error between two vectors. Given two vectors, x and y, the cosine error is defined as

$$L(x, y) = 1 - \cos(x, y),$$

where $cos(x, y)$ is the cosine of the angle between the vectors x and y.

Using this function causes a problem during prediction: the embedding network is not being optimized to match the original vectors' length. To compensate for this, when using an embedding network trained with this loss, we normalize the embedding vectors to the mean length of the original embedding matrix.

As an alternative to such normalization, we define loss functions that account for both direction and size, emphasizing direction. These combine the euclidean and cosine distances in a single loss function, one using addition and the other multiplication.

The additive euclidean-cosine error function is defined as:

$$L(x, y) = |x - y| + \alpha(1 - \cos(x, y)).$$

Whereas the multiplicative euclidean-cosine error is defined as:

$$L(x, y) = \alpha(|x - y|)(1 - cos(x, y)) + |x - y|.$$

Figure 5 depicts both functions when the target vector is $(1, 1)$. Notice that the second function should lead to faster

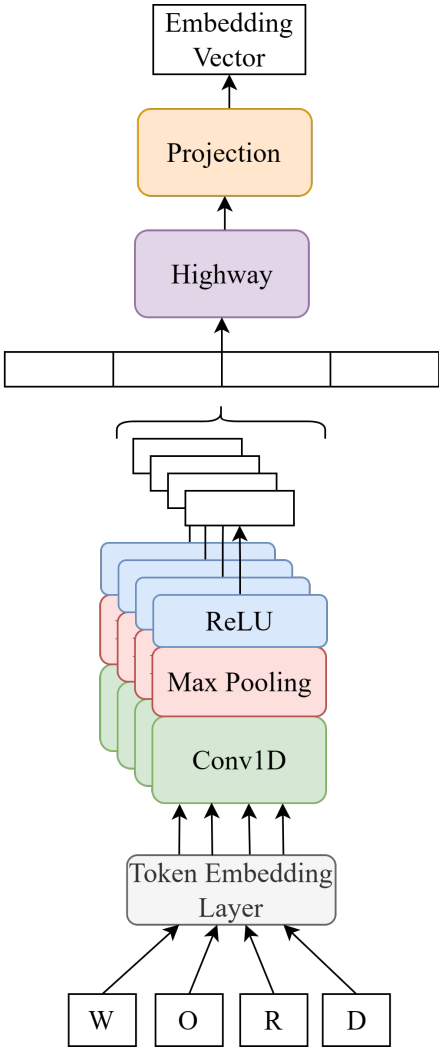

Fig. 3: BytesCNN architecture.

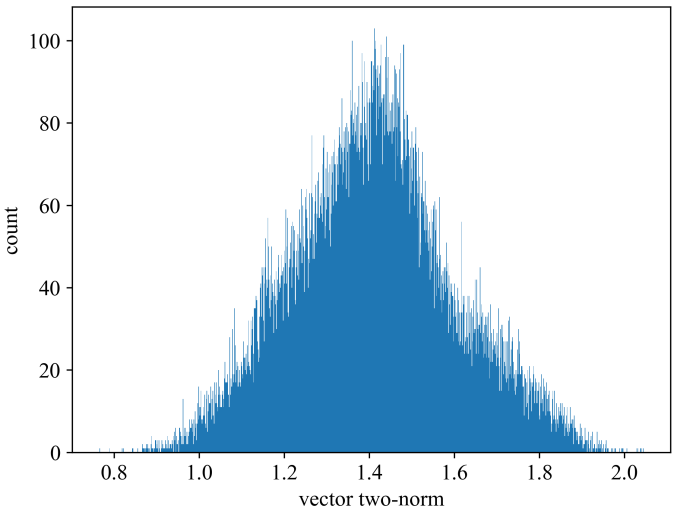

Fig. 4: Distribution of the norms of the vectors in the embedding layer of the BERT baseline model studied.

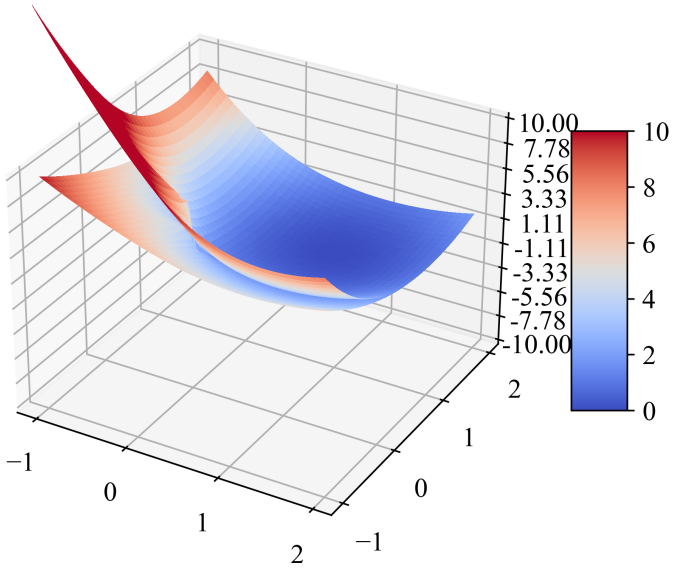

Fig. 5: Euclidean-cosine distance error functions when the target vector is $(1, 1)$. The surface that grows faster away from the $(1, 1)$ is the multiplicative euclidean-cosine function.

convergence to vectors in the same direction as the targets, as the error grows faster in the direction that diverges from the target.

C. Contrastive Learning

We also leverage training the aligned embedding layers with a contrastive learning target, similar to Mershad et al. [41]. Concretely, we use a triplet loss function. The triplet loss function aims to find the model that minimizes the euclidean distance between two similar vectors while maximizing the distance between unrelated vectors. Formally, the triplet loss is defined as:

$$L(x, y_p, y_n) = \max(|x - y_p|^2 - |x - y_n|^2 + \alpha, 0),$$

where x, y_p, y_n are referred to as an anchor vector, a positive vector, and a negative vector, respectively. For our purposes, the anchor vector is the output of the BytesCNN, the positive vector is the ground truth embedding given by the BERT word embedding layer, and the negative vector is selected following the same procedure as Mershad et al. [41].

Additionally, we incorporate a variant of triplet loss that uses the cosine error instead of the Euclidean distance. This formulation of the triplet loss has been explored before in the context of person re-identification [45], [46]. It is referred to as Angular Triplet Loss, and it is defined as follows:

$$L(x, y_p, y_n) = \max(\cos(x, y_n) - \cos(x, y_p) + \alpha, 0).$$

Notice the inversion of the positive and negative operations is a result of the definition of the cosine error between two vectors, x and y, as $1 - \cos(x, y)$.

For our experiments, the value of α in all the triplet loss evaluations was set to 1.

IV. Experimental Setup and Results

We use BERT [3] to evaluate our proposed methods. Although, conceptually, the method could be applied to any transformer architecture or any network that uses a word embedding layer. The base model in our experiments is `bert-base-uncased`, available at the Huggingface models hub.[1]

To train all our models, we used 10^4 epochs with a batch size of 100. We used the Adam optimizer with default parameters: $\alpha = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. Additionally, we reduced the learning rate on the plateau by a factor of 0.9.

The BERT vocabulary size is 30522 words. However, we removed from the training procedure all the unused tokens, i.e., those reserved for future additions, leaving a total of 29528 words.

We used two BytesCNN architectures that differ in size. We call them BytesCNN-small and BytesCNN-big. The small variant uses the same configuration defined by Boukkour et al. [34], seven 1D convolutional layers with the following filters: $(1, 32)$, $(2, 32)$, $(3, 64)$, $(4, 128)$, $(5, 256)$, $(6, 512)$ and $(7, 1024)$. In the filter denoted as (K, O), K represents the kernel size, and O is the number of output channels. The BytesCNN-big duplicates each filter. Table I shows a comparison of the number of parameters of each of these variants compared to the BERT word embedding layer, ignoring the parameters associated with the unused tokens.

TABLE I: Number of parameters of the different embedding layers.

Model	Num. parameters
BERT word embedding	22,677,504
BytesCNN-small	18,562,416
BytesCNN-big	70,674,288

We conducted our experiments in the two pre-training tasks of BERT: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). We consider the performance in these two tasks to be essential for they are what the baseline is pre-trained on.

For evaluation, we used the `wikitext-2-raw -v1` subset of Wikipedia (Wikitext dataset). We also ran the experiments on the IMDB dataset. In both cases, we used the train split. The data is available at the Huggingface datasets hub.[2] [3]

Table II shows the accuracy of each scenario as evaluated on MLM and NSP in the Wikitext and IMDB datasets. For each model size and each task, the top-3 performing models are in italics, and the best-performing is highlighted in bold.

As observed, none of the alternative models achieve exactly the same performance as the baseline. Since those are only trained to mimic the baseline, we do not expect a better performance from the student models. However, some of them obtain very close results to the baseline. To validate this, we conducted two-sample paired t-tests to determine whether there is a statistically significant difference between the baseline and each one of the models. Figure 6 depicts the resulting p-values and highlights two significance levels: 0.05 and 0.1.

[1]https://huggingface.co/bert-base-uncased

[2]https://huggingface.co/datasets/wikitext/viewer/wikitext-2-v1/train

[3]https://huggingface.co/datasets/imdb

TABLE II: Accuracy score of the BERT with a BytesCNN aligned embedding layer in two datasets on the MLM and NSP tasks. Baseline performance included for reference.

	Wikipedia		IMDB	
Scenario	**MLM**	**NSP**	**MLM**	**NSP**
BERT (baseline)	0.6226	0.9421	0.5748	0.6570
BytesCNN-small				
MSE	0.2666	0.6215	0.2059	0.5267
Cosine	**0.5600**	0.9364	**0.5022**	*0.6296*
Additive euclidean-cosine	*0.5271*	0.9369	*0.4692*	*0.6336*
Multiplicative euclidean-cosine	0.3071	0.7102	0.2504	0.5292
Triplet (euclidean)	0.0939	0.5603	0.0414	0.5129
Triplet (angular)	*0.5578*	0.9372	*0.5002*	**0.6358**
BytesCNN-big				
MSE	0.5876	0.9407	0.5319	0.6474
Cosine	*0.5919*	0.9397	0.5356	*0.6560*
Additive euclidean-cosine	0.5883	0.9416	***0.5374***	***0.6568***
Multiplicative euclidean-cosine	***0.5914***	0.9412	*0.5376*	0.6539
Triplet (euclidean)	0.3005	0.6119	0.1836	0.5390
Triplet (angular)	*0.5912*	0.9397	*0.5350*	*0.6544*

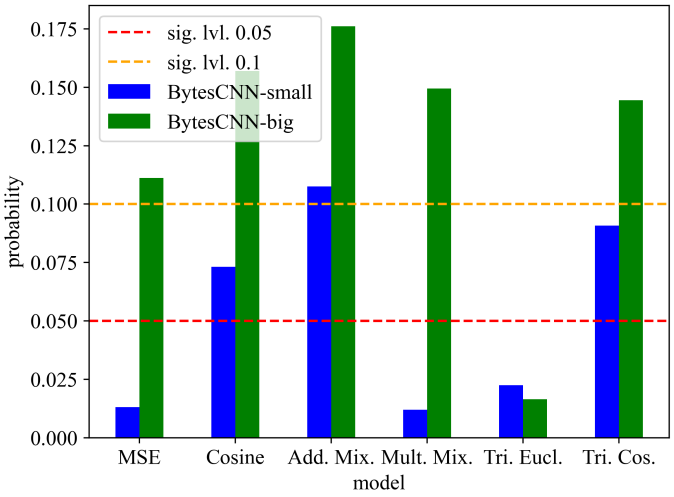

Fig. 6: Bar plots showing the p-values of several two-sample paired t-test. Each bar corresponds to a t-test where one sample is the baseline model and the other is the model labeled in the plot. Blue bars correspond to the BytesCNN-small model and the green ones to the BytesCNN-big. The dashed lines represent two significance levels considered.

Several conclusions can be drawn. First, the difference between small models and the respective big models is notable. All the big model variants' results (except for the Triplet Euclidean) do not provide significant evidence to reject the null hypothesis under any significance level displayed, meaning they are not significantly different from the baseline as far as these experiments are concerned. Additionally, note that models that were trained using a form of cosine-based distance perform better than those that solely use Euclidean-based losses.

Next, we briefly analyze the aligned embedding space, seen through the lenses of the BytesCNN-big model trained with the additive euclidean-cosine loss. Figure 7 depicts illustrations of

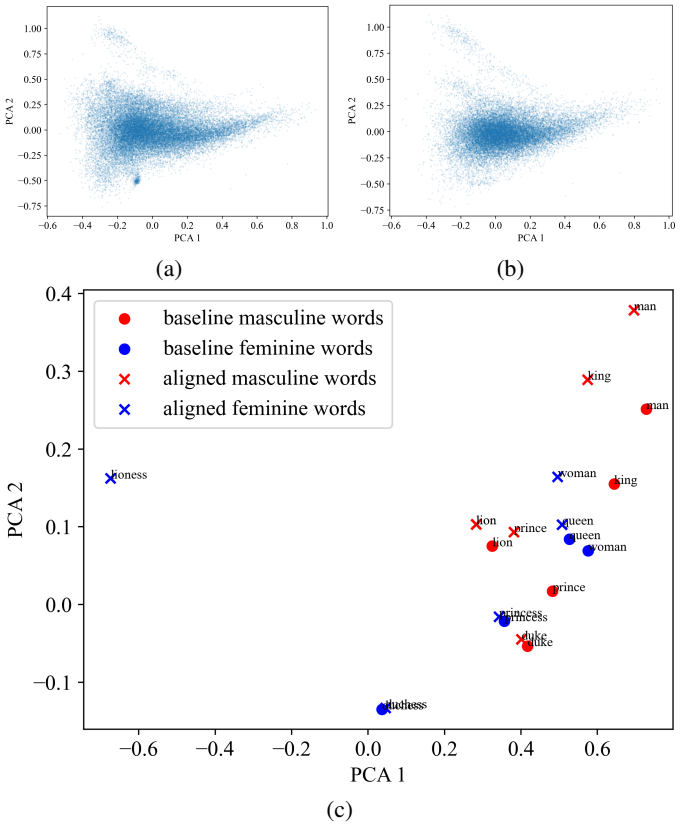

Fig. 7: Principal Component Analysis plots of the baseline and the aligned embedding spaces. 7a shows the entire vocabulary of the baseline embedding space. 7b shows the entire vocabulary of the aligned embedding space. We used the principal components learned for the baseline space for both plots. 7c depicts a selection of words in both spaces where there is a morphological distinction between masculine and feminine variants. For clarity, the paired words are: (*man, woman*), (*king, queen*), (*prince, princess*), (*duke, duchess*), (*lion, lioness*). Note the word *lioness* is not part of the baseline embedding vocabulary. Hence, only the vector in the aligned space is shown.

the vectors in the respective embedding spaces after applying Principal Component Analysis to reduce the dimensions of the vectors.

Notably, panels 7a and 7b display a significant resemblance between the two clouds of points. Second, note in panel 7c how the original and aligned versions of the words appear relatively close in the two-dimensional space. Importantly, for the words that appear in the baseline vocabulary, there is a pattern that relates the feminine version of a word with its masculine counterpart. However, the representation of the word *lioness* that the BytesCNN model comes up with, does not follow this pattern. This is clearly a limited analysis, yet we can observe that even though the BytesCNN embedding model is flexible enough to represent any sequence of bytes, it may not necessarily do so in a sensible way. Further studies are required to investigate how much fine-tuning would be required for the aligned model to develop a good representation of out-of-vocabulary words.

We also conducted experiments to test if using a BytesCNN as the word embedding layer makes the transformer more robust without any extra fine-tuning. We defined a simple model of noise that randomly replaces a percentage of the characters in the input sentence with a character also present in the testing data. Further investigation is required, but preliminary results show that the transformer with the BytesCNN embedding layer is not more robust against the tested model of noise than the baseline model.

V. Conclusion

In this paper we explored the possibility of transferring the knowledge of a vocabulary-fixed embedding layer into a CNN-based neural network that generates word representations based on bytes sequences. To that end, we followed a teacher-student-like methodology and studied a variety of loss functions to fit the student's network representation to the teacher's representation. Results show that it is feasible to align the bytes-based embedding to the baseline word embedding matrix, thus allowing to effectively convert a vocabulary-rigid model into a vocabulary-free one while preserving its knowledge to a great extent. Additionally, we showed empirically how cosine-based metrics can prove a better option to train the student network than euclidean-based loss functions are.

Our approach makes several important contributions to the field of NLP, including a method for re-training the embedding layer of transformer-based models, and an assessment of different loss functions for the alignment process. Our findings have important implications for the development of more flexible and robust NLP models that can handle a wide range of inputs, including those found in forensic applications. Future research could explore the application of our approach to other NLP tasks and investigate the potential of using byte-based representations to improve the forensic capacity of NLP models.

Limitations

We acknowledge the following limitations in the execution and results of this investigation.

First, while we argue that our method can be theoretically applied to any transformer-based model, we conducted experiments using only a BERT model as a baseline. Additionally, our evaluation is limited to the pre-training tasks of BERT, and we did not conduct a thorough hyperparameter study.

Second, our hypothesis regarding the distribution of embedding vectors' size in the vocabulary of BERT may not hold for other LLMs. As such, the success of the cosine-based model may be specific to BERT and may not generalize to other models.

Furthermore, our approach appears to degrade the generalization ability of transformer-based models, at least in the datasets and tests we conducted with statistical significance. However, additional research is necessary to draw definitive conclusions in this regard.

Finally, our objective was to distill the embedding layer, which would involve reducing the number of parameters. However, the best results are obtained using a larger model.

Overall, while our investigation provides promising results, further research is needed to address these limitations and extend the applicability of our method to other transformer-based models.

Ethics Statement

This research did not involve human subjects, so no human data was used. The research solely relied on pre-existing datasets and the BERT models. The model used in this research has been thoroughly tested on the aforementioned benchmark datasets, and we do not have any concerns about our model's ethical implications.

However, it is important to note that our model may inherit known concerns from BERT original embeddings. As the BERT embeddings come from a version of the model trained on a large corpus of text, there may be inherent biases in the data that are reflected in the embeddings. We acknowledge the potential ethical implications of these biases and emphasize the need for further research and exploration to mitigate these concerns. In our work, we attempted to show how we created and tested our model providing transparency in our methodology.

Acknowledgements

This article is based on work supported by the National Science Foundation under Grant No. 2039678, 2136961, and 2210091. Any opinions, findings, conclusions, or recommendations expressed in this article are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] K. T. Chitty-Venkata, M. Emani, V. Vishwanath, and A. K. Somani, "Neural architecture search for transformers: A survey," *IEEE Access*, vol. 10, pp. 108 374–108 412, 2022.

[2] Y. Bondarenko, M. Nagel, and T. Blankevoort, "Understanding and overcoming the challenges of efficient transformer quantization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 7947–7969. [Online]. Available: https://aclanthology.org/2021.emnlp-main.627

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[5] K. Bostrom and G. Durrett, "Byte pair encoding is suboptimal for language model pretraining," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4617–4624. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.414

[6] S. Klein and R. Tsarfaty, "Getting the ##life out of living: How adequate are word-pieces for modelling complex morphology?" in *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Online: Association for Computational Linguistics, Jul. 2020, pp. 204–209. [Online]. Available: https://aclanthology.org/2020.sigmorphon-1.24

[7] V. Hofmann, J. Pierrehumbert, and H. Schütze, "Superbizarre is not superb: Derivational morphology improves BERT's interpretation of complex words," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 3594–3608. [Online]. Available: https://aclanthology.org/2021.acl-long.279

[8] Y. Dong, S. Wang, Z. Gan, Y. Cheng, J. C. K. Cheung, and J. Liu, "Multi-fact correction in abstractive text summarization," *arXiv preprint arXiv:2010.02443*, 2020.

[9] K. Xu, H. Wu, L. Song, H. Zhang, L. Song, and D. Yu, "Conversational semantic role labeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2465–2475, 2021.

[10] M. Bromberg, L. Welmans, and C. Lee, "Reading between the text (s)-interpreting emoji and emoticons in the australian criminal law context," *New Criminal Law Review*, vol. 23, no. 4, pp. 655–686, 2020.

[11] J. Pei and L. Cheng, "Deciphering emoji variation in courts: a social semiotic perspective," *Humanities and Social Sciences Communications*, vol. 9, no. 1, pp. 1–8, 2022.

[12] E. Tong, A. Zadeh, C. Jones, and L.-P. Morency, "Combating human trafficking with multimodal deep models," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1547–1556.

[13] A. Wagner, S. Marusek, and W. Yu, "Emojis and law: contextualized flexibility of meaning in cyber communication," *Social Semiotics*, vol. 30, no. 3, pp. 396–414, 2020.

[14] L. Wang, E. Laber, Y. Saanchi, and S. Caltagirone, "Sex trafficking detection with ordinal regression neural networks," *ArXiv*, vol. abs/1908.05434, 2019.

[15] J. Zhu, L. Li, and C. Jones, "Identification and detection of human trafficking using language models," in *2019 European Intelligence and Security Informatics Conference (EISIC)*. IEEE, 2019, pp. 24–31.

[16] K. Cao and L. Rimell, "You should evaluate your language model on marginal likelihood over tokenisations," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2104–2114. [Online]. Available: https://aclanthology.org/2021.emnlp-main.161

[17] X. Wang, S. Ruder, and G. Neubig, "Multi-view subword regularization," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 473–482. [Online]. Available: https://aclanthology.org/2021.naacl-main.40

[18] V. Hofmann, H. Schuetze, and J. Pierrehumbert, "An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 385–393. [Online]. Available: https://aclanthology.org/2022.acl-short.43

[19] W. Ma, Y. Cui, C. Si, T. Liu, S. Wang, and G. Hu, "CharBERT: Character-aware pre-trained language model," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 39–50. [Online]. Available: https://aclanthology.org/2020.coling-main.4

[20] N. Banar, W. Daelemans, and M. Kestemont, "Character-level transformer-based neural machine translation," in *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, ser. NLPIR 2020. New York, NY, USA: Association for Computing Machinery, 2021, p. 149–156. [Online]. Available: https://doi.org/10.1145/3443279.3443310

[21] Y. Pinter, A. Stent, M. Dredze, and J. Eisenstein, "Learning to look inside: Augmenting token-based encoders with character-level information," *ArXiv*, vol. abs/2108.00391, 2021.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and

K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

[24] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[25] O. Melamud, J. Goldberger, and I. Dagan, "context2vec: Learning generic context embedding with bidirectional lstm," in *Proceedings of the 20th SIGNLL conference on computational natural language learning*, 2016, pp. 51–61.

[26] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," *Advances in neural information processing systems*, vol. 30, 2017.

[27] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)," pp. 2227–2237, 2018.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[29] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[30] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, vol. abs/1907.11692, 2019.

[31] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=H1eA7AEtvS

[32] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703

[33] S. J. Mielke, Z. Alyafeai, E. Salesky, C. Raffel, M. Dey, M. Gallé, A. Raja, C. Si, W. Y. Lee, B. Sagot, and S. Tan, "Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp," *ArXiv*, vol. abs/2112.10508, 2021.

[34] H. El Boukkouri, O. Ferret, T. Lavergne, H. Noji, P. Zweigenbaum, and J. Tsujii, "CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 6903–6915. [Online]. Available: https://aclanthology.org/2020.coling-main.609

[35] J. H. Clark, D. Garrette, I. Turc, and J. Wieting, "Canine: Pre-training an efficient tokenization-free encoder for language representation," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 73–91, 2022. [Online]. Available: https://aclanthology.org/2022.tacl-1.5

[36] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, "Byt5: Towards a token-free future with pre-trained byte-to-byte models," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 291–306, 2022.

[37] C. Lee, Q. Guo, and X. Qiu, "Word-level representation from bytes for language modeling," *ArXiv*, vol. abs/2211.12677, 2022.

[38] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv*, vol. abs/1503.02531, 2015.

[39] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[40] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[41] A. Mersha and W. Stephen, "Distilling word embeddings via contrastive learning," *Transfer Learning for NLP Workshop 2022 – WiNLP*, 2022.

[42] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[43] Y. Tay, V. Q. Tran, S. Ruder, J. Gupta, H. W. Chung, D. Bahri, Z. Qin, S. Baumgartner, C. Yu, and D. Metzler, "Charformer: Fast character transformers via gradient-based subword tokenization," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=JtBRnrlOEFN

[44] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *ArXiv*, vol. abs/1505.00387, 2015.

[45] H. Ye, H. Liu, F. Meng, and X. Li, "Bi-directional exponential angular triplet loss for rgb-infrared person re-identification," *IEEE Transactions on Image Processing*, vol. 30, pp. 1583–1595, 2020.

[46] Y. Li, R. Xue, M. Zhu, J. Xu, and Z. Xu, "Angular triplet loss-based camera network for reid," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–7.