The Unreasonable Effectiveness of Random Target Embeddings for Continuous-Output Neural Machine Translation

Evgeniia Tokarchuk **Vlad Niculae**
Language Technology Lab
University of Amsterdam
{e.tokarchuk, v.niculae}@uva.nl

Abstract

Continuous-output neural machine translation (CoNMT) replaces the discrete next-word prediction problem with an embedding prediction. The semantic structure of the target embedding space (*i.e.*, closeness of related words) is intuitively believed to be crucial. We challenge this assumption and show that completely random output embeddings can outperform laboriously pre-trained ones, especially on larger datasets. Further investigation shows this surprising effect is strongest for rare words, due to the geometry of their embeddings. We shed further light on this finding by designing a mixed strategy that combines random and pre-trained embeddings, and that performs best overall.

1 Introduction

Since text is naturally discrete, *i.e.*, each token in a target sentence is represented by an integer index in the vocabulary, neural machine translation (NMT), as many other language generation tasks, is trained mainly as a discrete-output model with softmax over the full vocabulary followed by the cross-entropy loss. Continuous-output neural machine translation (CoNMT) models, in contrast, are trained to predict the continuous representation based on the distances between vectors. It is an appealing line of study for computational and modeling related reasons (Kumar and Tsvetkov, 2019), as well as a reliable test bed for exploring the properties of continuous language spaces that appear in modern deep generative models (Li et al., 2022b). However, CoNMT introduces its own challenge, namely mapping to and from a continuous space. During training, CoNMT model requires continuous targets, and while decoding, one needs to map back to the discrete text representation.

Text mapping to continuous space is widely explored in NLP and can be done using *embeddings* of tokens, words (Turian et al., 2010; Mikolov et al., 2013, 2018) and sentences (Reimers and Gurevych, 2019; Feng et al., 2022). Cosine similarity between word embeddings is well correlated with lexical similarity metrics, motivating the use of cosine distance against pre-trained embeddings as an effective training strategy for CoNMT Nearest neighbor beam decoding would in this case include related words and, unlike discrete cross-entropy, the training strategy does not discourage synonyms.

Previous studies show that the quality of continuous-output models highly depends on the choice of embeddings (Li et al., 2022b; Tokarchuk and Niculae, 2022; Kumar and Tsvetkov, 2019). In general, in CoNMT the embeddings are pre-trained and fixed: otherwise, making all embeddings equal yields an unwanted global optimum. Obtaining pre-trained word embeddings can be computationally expensive, especially if one needs to train an embeddings model from scratch.

In this work we *randomly* initialize target embeddings for continuous-output models and keep them static during training. Arora et al. (2020) applied static random embeddings for text classification model's input; however, to the best of our knowledge, the effect of untrained random target embeddings has not been previously studied in the literature, especially for text-generating tasks such as machine translation. However, we show that random target embeddings perform close to their pre-trained counterpart, and even surpass them on the larger datasets, challenging the assumption that target embeddings must preserve semantic relationships. Meaningful structures in target embedding space could help with generalization, but our results suggest that any such benefits are smaller than one might expect, and sensitive to embedding concentration. We hypothesize and bring experimental evidence that CoNMT performance is negatively impacted when there is too little space around embeddings, *i.e.*, when embeddings are tangled rather than more spread out. Our findings on three NMT tasks, namely WMT 2018 English→Turkish

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 653–662

(en-tr), WMT 2016 English→Romanian (en-ro), and WMT 2019 English→German (en-de) indicate that random embeddings are more spread out and perform better on rare words for all language pairs. Strikingly, on the largest dataset (en-de), random embeddings show the largest gain over pre-trained ones. We propose a simple yet efficient combination of random and pre-trained embeddings, and show that it improves model performance in most cases considered. More generally, our findings show that dispersion is an important property of embedding space geometry, and that integrating semantic information should be done with care.

2 Continuous-Output NMT

The machine translation task involves learning to map sequences of input tokens $\boldsymbol{x} = (x_1, \ldots, x_m)$ to output tokens $\boldsymbol{y} = (y_1, \ldots, y_n)$. In standard (discrete) NMT, each step is a multi-class next word prediction task, minimizing:

$$\begin{aligned} L_{\text{discrete}}(y_i = t; \boldsymbol{y}_{<i}, \boldsymbol{x}) &= -\log p(y_i = t \mid \boldsymbol{y}_{<i}, \boldsymbol{x}) \\ &= -\langle \boldsymbol{E}(t), \boldsymbol{h}\rangle + \log \sum_{t' \in V} \exp \langle \boldsymbol{E}(t'), \boldsymbol{h}\rangle, \end{aligned} \tag{1}$$

where t is a token index, V is the vocabulary, $\boldsymbol{E} : V \longrightarrow \mathbb{R}^d$ is an embedding lookup, and $\boldsymbol{h}$ is a transformer hidden state calculated in terms of $\boldsymbol{x}$ and the output prefix $\boldsymbol{y}_{<i}$. The costly log-sum-exp and the penchant for continuous similarity metrics in NLP motivate a purely-continuous alternative:

$$L_{\cos}(y_i = t; \boldsymbol{y}_{<i}, \boldsymbol{x}) = 1 - \cos(\boldsymbol{E}(t), \boldsymbol{h}). \tag{2}$$

Continuous NMT models were first studied by Kumar and Tsvetkov (2019), who also propose other probabilistic losses and later other margin-based objectives (Bhat et al., 2019), with limited gain and at the cost of additional hyperparameters; we therefore focus on the robust cosine objective. We further justify the choice of cosine over max-margin as an objective function in Appendix C.1.

On the other hand, the choice of embeddings $\boldsymbol{E}$ makes a much larger difference, especially due to the fact that all previous work keeps this parameter frozen: indeed, if it were trainable, Equation (2) would have trivial global optima by setting all $\boldsymbol{E}(t)$ to the same vector for all t. With modern transformer architectures, the best performing embeddings overall tend to be the "oracle" output embeddings learned by a pre-trained discrete MT system (Tokarchuk and Niculae, 2022). We highlight that the cosine loss is invariant to the norms of both the embeddings and of the decoder hidden state, and therefore we may restrict our modeling problem to the unit sphere.

Optimizing Equation (1) pushes the model $\boldsymbol{h}$ away from all tokens different from the "gold" token, even if some other tokens (*e.g.*, synonyms) could otherwise be a good fit. Equation (2) has no such effect, leading to a promise of more diverse generations. An appealing intuition is that synonyms and related words being nearby in embedding space contributes to the performance of CoNMT and enables such diversity. However in practice, greedy nearest-neighbor lookup is applied, and beam search decoding is not well-studied in the context of CoNMT. Therefore, in this work, we dwell more into the beam search performance for CoNMT, and compare pre-trained and completely random embeddings.

3 Random Embeddings Generation

We consider two different distributions from which to sample the $|V|$ random embeddings.

Spherical uniform. We draw embeddings uniformly from the surface of the sphere: $\boldsymbol{E}(y_i) \sim \text{Unif}(\mathbb{S}_{d-1})$. Since standard normal vectors are distributed with rotational symmetry around the origin, uniform samples on the sphere can be obtained by normalizing standard normal random vectors:

$$\boldsymbol{E}(y_i) = \boldsymbol{u}_i / \|\boldsymbol{u}_i\|; \quad \boldsymbol{u}_i \sim \text{Normal}(\boldsymbol{0}, \boldsymbol{I}_d).$$

The same argument works if the normal distribution has spherical covariance $\sigma \boldsymbol{I}_d$ for any σ, and thus, since the cosine loss is norm-invariant, uniform initialization is exactly equivalent to the standard initialization of transformer embeddings.

Hypercube. The corners of the hypercube $\{-1, 1\}^d$ all have norm $\sqrt{d}$ and thus form a discrete subset of a hypersphere. This motivates us to consider drawing embeddings from a scaled Rademacher distribution:

$$\boldsymbol{E}(y_i) = \boldsymbol{r}_i / \sqrt{d}; \quad \boldsymbol{r}_i \sim \text{Rademacher}(d).$$

Each coordinate of $\mathbf{r}_i$ has 50% probability of being +1 and 50% of being −1. With this strategy, any two distinct embeddings have cosine distance at least $2/d$. Moreover, hypercubic embeddings can be stored as bit patterns and potentially allow for faster loss calculation with dedicated low-level implementations which we do not explore here.

embeddings	en-tr BLEU ↑	en-tr BERTSc. ↑	ro-en BLEU ↑	ro-en BERTSc. ↑	en-de BLEU ↑	en-de BERTSc. ↑
discrete model	12.3	70.4	31.7	64.1	33.1	69.0
pre-trained (beam=1)	10.1	67.1	29.0	58.5	31.3	66.2
pre-trained	**10.4**	67.4	29.0	58.0	29.2	62.6
random uniform	8.9	65.1	28.8	58.8	31.8	**67.2**
random cube	8.7	64.6	28.7	58.8	31.4	66.9
combined	**10.4**	**68.3**	**29.5**	**60.4**	**32.0**	66.8

Table 1: BLEU and BertScore on `ro-en` `newstest16`, `en-tr` `newstest2017` and `newstest2016` `en-de`. We use a beam of 5 if not stated otherwise. In bold, we show the highest score among the continuous models in each column.

4 Experimental Setup and Data

We train CoNMT systems with pre-trained target embeddings as well as randomly-generated target embeddings. The **pre-trained embeddings** we use are extracted from a discrete NMT system trained on the same training data, following the setup of Tokarchuk and Niculae (2022), who found this strategy to outperform other subword-level pre-trained embeddings for CoNMT.

Results are reported on three WMT translation tasks:[1] WMT 2016 Romanian→English (`ro-en`), WMT 2018 English→Turkish (`en-tr`) and WMT 2019 English→German (`en-de`), the latter including back-translated data. Note that for `en-tr` we use only WMT 2018 training data with 207k training sentences in order to investigae a challenging lower-resource and morphology-rich scenario. Data statistics are collected in Appendix A.

For subword tokenization we used the same SentencePiece (Kudo and Richardson, 2018) model for all language pairs, specifically the one used in the mBart multilingual model (Liu et al., 2020). This choice allows for unified preprocessing for all languages we cover. We validate that token-based models performs generally better than word-level models (Appendix C.4), even though subwords introduce an additional challenge of predicting subword continuation (Appendix C.5).

We used the `fairseq` (Ott et al., 2019) framework for training our models. Baseline discrete models are trained with cross-entropy loss, label smoothing equal to 0.1 and effective batch size 65.5K tokens. Both discrete and continuous models are trained with learning rate $5 \cdot 10^{-4}$, 10k warm-up steps for `ro-en` and `en-de`, and 4k for the smaller `en-tr` dataset. All continuous models are trained with the cosine distance objective in Equation (2). We provide all training details in Appendix B.

[1]https://www2.statmt.org/

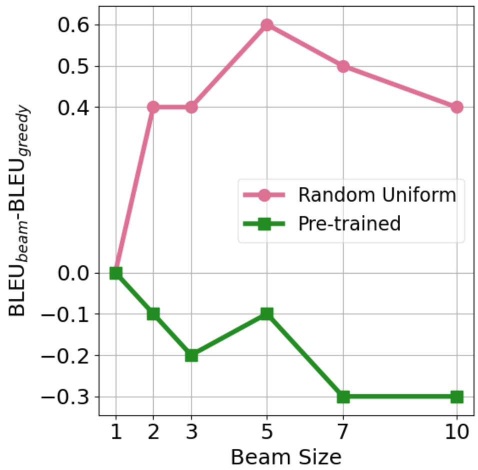

Figure 1: BLEU$_{\text{beam}}$−BLEU$_{\text{greedy}}$ scores for the `ro-en` `newsdev2016` for continuous output models with uniform random and pre-trained embeddings. Greedy (beam size 1) BLEU scores are 30.0 for pre-trained, and 28.6 for random embeddings.

We measure translation accuracy using SacreBLEU[2] (Papineni et al., 2002; Post, 2018) and BertScore[3] (Zhang et al., 2020). Note that BertScore is scaled differently for each language, so the scores cannot be compared across languages.

5 Results and Discussion

Scores. Per Table 1, we find that random uniform embeddings outperform the pre-trained baseline for `en-de`, match it closely for `ro-en`, and only underperform in the low-resource case for `en-tr`. We find that hypercube embeddings consistently perform no better than uniform embeddings; however, it is possible that their computational advantages can make up for this in some applications.

Beam search. Preliminary experiments with CoNMT models indicate little gain or even degra-

[2]nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1
[3]implementation by https://github.com/Tiiiger/bert_score

dation from beam search, which is why we also report results with greedy decoding for pre-trained in Table 1. Further investigation in Figure 1 shows that the ro-en model with pre-trained embeddings degrades consistently, performing best in the greedy case, while the random embedding model benefits noticeably from a larger beam, in spite of neighboring words being random and not related. We discuss the details of the beam search in Appendix D.

Frequency. We perform a token-level evaluation using compare-mt (Neubig et al., 2019), computing the F_1 score of matching a gold token (at its gold position), aggregated over bins defined by the token's frequency in the training data. The result in Figure 3 reveals that random embeddings allow much better classification of rare tokens than even the discrete reference model. To understand this effect, we study the geometry of the pre-trained embedding spaces in relation to frequency in Figure 2. The top row shows the relationship between the frequency *rank* (higher means rarer) and the similarity to its nearest– and fifth-nearest– neighbors. For all three language pairs we observe that most rare words become identical to their nearest neighbor. In contrast, for random embeddings this metric does not depend on rank and is always around 0.4. The bottom row of Figure 3 shows that the nearest neighbors of rare words tend also to be comparably rare. This geometry clarifies in part the surprising performance of random embeddings on rare tokens.

Combined embeddings. Our finding motivates combining pre-trained and random embeddings:

$$\boldsymbol{E}_{\text{cmb}}(y_i) = \frac{\alpha \boldsymbol{E}_{\text{pre}}(y_i) + (1-\alpha)\boldsymbol{E}_{\text{rand}}(y_i)}{\|\alpha \boldsymbol{E}_{\text{pre}}(y_i) + (1-\alpha)\boldsymbol{E}_{\text{rand}}(y_i)\|}.$$

To emphasize pre-trained distances more than the noise, we choose $\alpha = 0.9$ for all language pairs. This simple approach leads to overall improved performance, on almost all metrics and language pairs as shown in Table 1. Furthermore, Figure 3 confirms that combined embeddings preserve the performance of pre-trained embeddings on frequent tokens and increase F_1 score on rare tokens. We further study the impact α on ro-en in Appendix C.3 and observe that for all considered $\alpha \in [0.5, 0.9]$, the combination outperforms random and pre-trained embeddings along both metrics; the specific value of α in this range has only negligible impact.

6 Additional Related Work

CoNMT losses. Earlier work in CoNMT suggests loss functions other than cosine, based on modified Langevin (a.k.a. von Mises-Fisher) log-likelihood, or based on max-margin constructions, to perform better (Kumar and Tsvetkov, 2019; Bhat et al., 2019). Nevertheless, in preliminary experiments, we find that when using more modern architectures and datasets, such objectives do not outperform the cosine loss. The cosine loss is an instance of Langevin log-likelihood with spread $\kappa = 1$ (Appendix D), allowing for a theoretically-grounded beam search over sequence likelihood, whereas for max-margin losses it is not clear how to derive a principled beam search. Nevertheless, we provide a small set of additional experiments confirming that max-margin losses underperform cosine while showing similar effects in Appendix C.1.

Retrieval-augmented NMT. Similarly to CoNMT, k-NN MT (Khandelwal et al., 2021; Yogatama et al., 2021; Stap and Monz, 2023) relies on the distance-based retrieval from datastore in decoding time, with cosine similarity and Euclidean distance as a popular choice of the similarity measure. Even though creation of a datastore and extracting target embeddings are two distinct processes, they both share similar traits and rely on discrete transformer MT system as a source of representations. Li et al. (2022a) argue that quality of k-NN MT directly depends on the quality of retrieved neighbors contexts from the datastore, and show that k-NN MT exhibits a related issue with high similarity between unrelated keys. Our findings suggests that randomization could provide paths toward improved performance in k-NN MT.

Unargmaxability. Grivas et al. (2022) point out that standard (discrete) language models can have "unargmaxable" vocabulary items. When using directional modelling (on the unit sphere), unargmaxability is mitigated and only occurs for identical embeddings; however, embeddings that are too close to their neighbors can have very small Voronoi sets, leading to the phenomenon we identify in this work, which is problematic in practice for CoNMT. Random perturbations to embeddings might effectively mitigate unargmaxability in discrete models as well.

Hubness. Hubness (Dinu and Baroni, 2014; Lazaridou et al., 2015; Huang et al., 2019) is a phenomenon that impacts nearest-neighbor retrieval

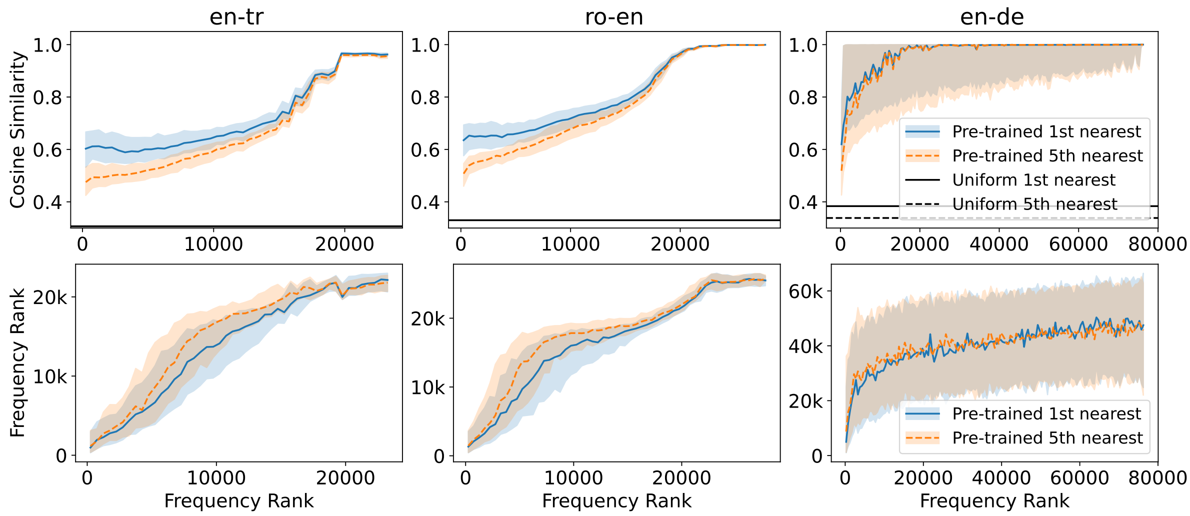

Figure 2: Pre-trained embeddings demonstrate strong correlation between the frequency rank of each token and (top) the cosine similarity, and (bottom) the frequency rank of its nearby neighbors. Most rare words are identified with their nearest neighbor, which is also a rare word. Bin size 500; shaded area denotes 50% of values in each bin.

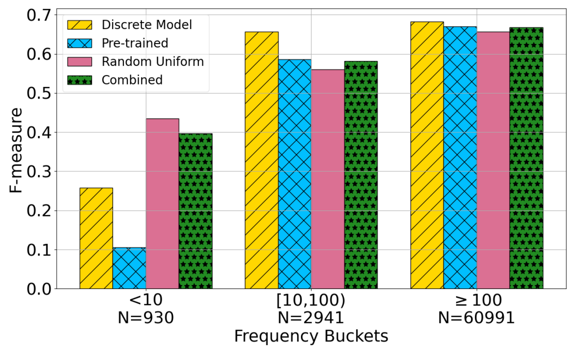

Figure 3: Token-level F_1 test score grouped into three bins defined by training set frequency. The x label shows frequency boundaries and token counts per bucket.

as well, characterized by the presence of a few data points (hubs) that are close to many other data points despite their semantic dissimilarity. The phenomenon we observe is related but different: many rare words are embedded very close to another rare word, but not necessarily close to all others overall. Therefore, methods for reducing hubness would not necessarily prevent this situation.

7 Conclusion

Our experimental results show that randomly initialized target embeddings can achieve similar performance as pre-trained ones and even surpass them when a sufficiently large amount of data is available. The gap is most pronounced on very rare tokens. We also found that beam size > 1 does not harm the performance of CoNMT with random target embeddings (compared to pre-trained target embeddings). We suggest combining random and pre-trained embeddings in attempt to maintain high accuracy on frequent tokens as well as rare tokens. This simple approach proved to be effective for `en-tr` and `ro-en` in terms of overall performance. However, more refined ways to combine random embeddings with semantically meaningful anchors may lead to more reliable improvements, and ideally hold the potential to remove the reliance on a pre-trained model entirely. Finding the best ways to achieve this potential is an important avenue of future work for CoNMT and for continuous modeling of language repesentations more broadly.

Limitations

Generalization. Our experimental results show that semantic similarity of the targets embeddings does not play a major role for continuous-output NMT. However, this may not necessarily hold for other text generation tasks like summarization or language modeling. To claim that random target embeddings can be sucessfuly used for any text generation task yet has to be proved. In the future, we will conduct additional experiments on other text generation tasks, such as summarization and language modeling.

Dataset Size. Arora et al. (2020) argue that random embeddings can achieve comparable performance when the dataset size is big enough. In our work we report results on three language pairs with vast range of training samples.The gap between pre-trained and random embeddings is much higher for `en-tr` with 207K training samples than for `ro-en` and `en-de` with 612K and 9.1M training samples. Moreover, on `en-de` random embeddings even outperform pre-trained ones. That hints that random embeddings indeed work only if there is sufficiently large amount of data available.

Static Embeddings. The formulation of the loss we use in our work, specifically cosine distance, leads to representation collapse when tuning target embeddings jointly with the model, That is why in our work the target embeddings are kept unchanged during training. Li et al. (2022b) show that it is possible to design a loss that allows for joint training. However, we believe that fine-tuning of random embeddings is orthogonal to our study.

Comparison with External Embeddings. In the scope of this work, we compared only embeddings extracted from the discrete NMT model (pre-trained) and randomly generated embeddings. However, we do not compare random embeddings with external models like mBart (Liu et al., 2020) or fasttext (Bojanowski et al., 2017). That is intentional since Tokarchuk and Niculae (2022) showed that pre-trained embeddings extracted from discrete NMT system perform the best compared to the external models, and our goal was to compare to the best-performing baseline.

Loss Function. All our results are tied to the choice of the target objective function, precisely cosine similarity. We chose cosine similarity to align our work with previous studies on CoNMT (Kumar and Tsvetkov, 2019; Tokarchuk and Niculae, 2022). Although our preliminary experiments with Langevin-based as well as with margin-based losses suggested worse performance than cosine for CoNMT, other less-studied objectives, *e.g.*, based on geodesic distances, or on expectations of a discrete loss (Scott et al., 2021), left outside of our scope, may lead to further improvement.

Risks

NMT as a technology is subject to dual-use concerns. We also want to stress that it is possible that random embedding models make different kinds of mistakes compared to other models, and they should be studied and treated with caution before deployment. CoNMT models are generally at an earlier stage of development and do not seem likely to replace the well-studied discrete models in deployed application in the very near future.

Acknowledgements

We thank all the members of the UvA Language Technology Lab for their valuable feedback on our work. Special thanks to Kata Naszádi, Sergey Troshin, Caio F. Corro, Andreas Grivas, and Wilker Aziz for their time and insightful comments. We also thank SURF (www.surf.nl) for the support in using the National Supercomputer Snellius. This work was partly supported by the Dutch Research Council (NWO) via VI.Veni.212.228 and the European Union's Horizon Europe research and innovation programme via UTTER 101070631.

References

Simran Arora, Avner May, Jian Zhang, and Christopher Ré. 2020. Contextual embeddings: When are they worth it? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2663, Online. Association for Computational Linguistics.

Gayatri Bhat, Sachin Kumar, and Yulia Tsvetkov. 2019. A margin-based loss with synthetic negative samples for continuous-output machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 199–205, Hong Kong. Association for Computational Linguistics.

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

Georgiana Dinu and Marco Baroni. 2014. Improving zero-shot learning by mitigating the hubness problem. *CoRR*, arXiv:1412.6568. Version 3.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Andreas Grivas, Nikolay Bogoychev, and Adam Lopez. 2022. Low-rank softmax can have unargmaxable classes in theory but rarely in practice. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6738–6758, Dublin, Ireland. Association for Computational Linguistics.

Jiaji Huang, Qiang Qiu, and Kenneth Church. 2019. Hubless nearest neighbor search for bilingual lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4072–4080, Florence, Italy. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Sachin Kumar and Yulia Tsvetkov. 2019. Von Mises-Fisher loss for training sequence to sequence models with continuous outputs. In *International Conference on Learning Representations*.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 270–280, Beijing, China. Association for Computational Linguistics.

Jiahuan Li, Shanbo Cheng, Zewei Sun, Mingxuan Wang, and Shujian Huang. 2022a. Better datastore, better translation: Generating datastores from pre-trained models for nearest neural machine translation. *CoRR*, arXiv:2212.08822. Version 1.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. 2022b. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. 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*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, Xinyi Wang, and John Wieting. 2019. compare-mt: A tool for holistic comparison of language generation systems. *CoRR*, abs/1903.07926.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Tyler R. Scott, Andrew C. Gallagher, and Michael C. Mozer. 2021. von Mises–Fisher loss: An exploration of embedding geometries for supervised learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10592–10602.

David Stap and Christof Monz. 2023. Multilingual k-nearest-neighbor machine translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9200–9208, Singapore. Association for Computational Linguistics.

Evgeniia Tokarchuk and Vlad Niculae. 2022. On target representation in continuous-output neural machine translation. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 227–235, Dublin, Ireland. Association for Computational Linguistics.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz

Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Dani Yogatama, Cyprien de Masson d'Autume, and Lingpeng Kong. 2021. Adaptive semiparametric language models. *Transactions of the Association for Computational Linguistics*, 9:362–373.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Kaitlyn Zhou, Kawin Ethayarajh, Dallas Card, and Dan Jurafsky. 2022. Problems with cosine as a measure of embedding similarity for high frequency words. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 401–423, Dublin, Ireland. Association for Computational Linguistics.

A Data Statistics

Table 2 contains data statistics for datasets used in our experiments.

B Models' Training Parameters

We report `fairseq` yaml config in Listing 1. Language-pair–specific parameters are highlighted with a comment. Continuous transformer uses base Transformer architecture with 6 layers of encoder and decoder (Vaswani et al., 2017). Total number of training parameters is the following: `ro-en` discrete is 42M and `ro-en` continuous 74M; `en-tr` discrete is 40M and `en-tr` continuous 73M; `en-de` discrete is 132M and `en-de` continuous 123M.

We train our models using shared GPU cluster, which is equipped with GeForce GTX TITAN X as well as NVIDIA A100.

C Additional Experiments

C.1 Max-Margin Loss

We experimented with two variants of max-margin loss described in Bhat et al. (2019), namely Syn-margin by projection (SMP) and Syn-margin by difference (SMD) on the `en-ro` dataset. Using the same hyperparameters as for cosine and discrete models (α=1, learning rate of 10^{-4}, and effective batch size of 65536) all max-margin models obtained scores below the best cosine model. Table 3 shows comparison of the models' performance when using max-margin loss and cosine loss for training CoNMT on `newstest2016 ro-en`. While these results may improve with tuning, it seems

Listing 1 Training yaml config for CoNMT.

```yaml
task:
  _name: translation
  data: language_specific_data
criterion:
  _name: cosine_ar_criterion
model:
  _name: continuous_transformer
  decoder:
    output_dim: 128
    learned_pos: true
  encoder:
    learned_pos: true
  dropout:
    0.3  # ro-en and en-tr
    0.1  # en-de
  target_embed_path: $PATH
  no_decoder_final_norm: false
optimizer:
  _name: adam
  adam_betas: (0.9,0.98)
lr_scheduler:
  _name: inverse_sqrt
  warmup_updates:
    10000  # ro-en and en-de
    4000   # en-tr
  warmup_init_lr: 1e-07
dataset:
  validate_after_updates: 10000
  max_tokens: 4096
  validate_interval_updates: 2000
optimization:
  lr: [0.0005]
  update_freq: [16]
  max_update: 50000
  stop_min_lr: 1e-09
checkpoint:
  no_epoch_checkpoints: true
  best_checkpoint_metric: bleu
  maximize_best_checkpoint_metric: true
```

	WMT ro-en			WMT en-tr				WMT en-de			
	train	dev16	test16	train	dev17	test17	test18	train	valid	test16	test18
sentences	612K	2K	2K	207K	1K	3K	3K	9.1M	2.2K	3K	3K
SPM vocabulary (tgt)	27.5K			23.3K				76K			
SPM % oov (tgt)	0.0	0.38	0.31	0.0	0.45	0.53	0.55	0.0	0.0	0.0	0.0

Table 2: Datasets Statistics

model	BLEU	BERT
cosine pre-trained (beam=1)	29.0	58.5
cosine pre-trained (beam=5)	29.0	58.0
cosine random (beam=1)	28.0	58.2
cosine random (beam=5)	28.8	58.8
SMP pre-trained	27.1	54.7
SMD pre-trained	28.5	57.5
SMP random	16.7	36.7
SMD random	26.3	54.3

Table 3: Compariosn between cosine and max-margin loss for `newstest2016` `ro-en`.

unlikely for the effect to be more important than the embedding choice, and our finding that random embeddings are at least competitive with pre-trained ones holds. The cosine loss remains a performant, simple, and robust training objective for CoNMT with a probabilistic interpretation, making it suitable for principled beam search, and thus we restrict the scope of our experiments to it.

C.2 Embeddings Dimensionality

Even though it is typical to train NLP models with large embeddings dimension ($d \geq 512$), we conducted experiments on `ro-en` and found that smaller dimensionality works better for CoNMT both with random and pre-trained target embeddings Figure 5, and do not harm the performance of discrete model as per Figure 4.

We hypothesise that better performance of lower dimensional embeddings on CoNMT is a direct consequences of the cosine distance as a distance measure. Despite its popularity, there is evidence that cosine loss is not a suitable choice for measuring the dissimilarity between high-dimensional embeddings vectors (Zhou et al., 2022), and using another distance metric can potentially improve the results of the models with larger embeddings dimensionality. We leave this question for the future investigation. Since the dimensionality 128 performs the best among all tested dimensionalities,

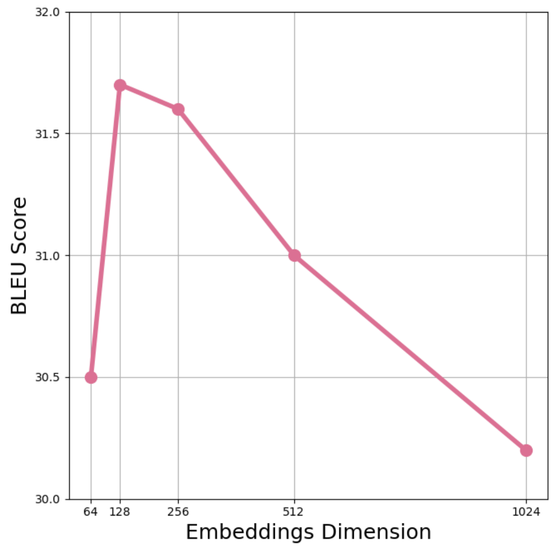

Figure 4: BLEU score of the discrete NMT models on `newstest2016` `ro-en`.

we do all our experiments with dimension equal to 128.

C.3 Combined Embeddings

In Table 1 we report performance of `combined` embeddings with $\alpha = 0.9$. To study the effect of α on the models' performance, we conduct experiments on `ro-en` for $\alpha \in [0.5, 0.9]$. As shown in Figure 6, for all cases `combined` embeddings outperform pre-trained and random ones on both metrics.

C.4 Word Embeddings for CoNMT

Since the continuous-output model struggles with subwords continuation and, at the same time, performs better on rare words, we conduct experiments on the word level. Word-level model tends to suffer from out-of-vocabulary issues (Table 2), so discrete model performance drops respectively. Table 4 provides the comparison between the discrete word-level model and continuous-output model with random targets. Even though the continuous-output model struggles with subwords continuations, overall, using subwords allows us to have a

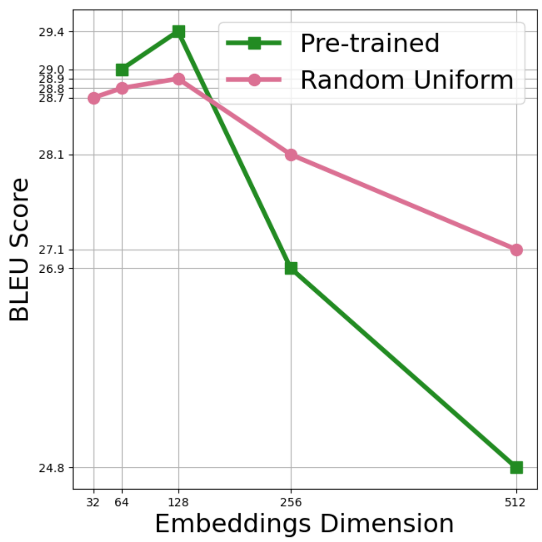

Figure 5: BLEU score on `ro-en` `newstest2016` of continuous-output model with various dimensionalities of random and pre-trained target embeddings.

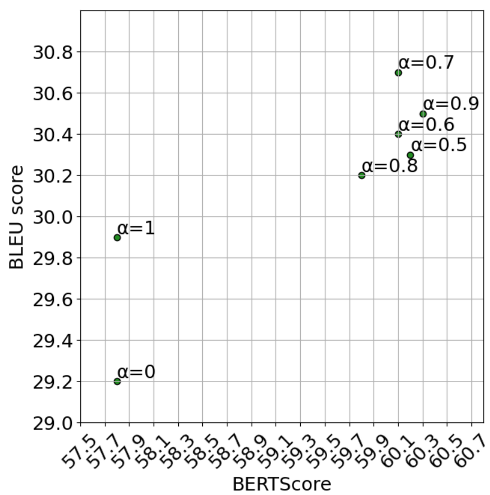

Figure 6: BLEU and BERTScores on `ro-en` `newsdev2016` with different values of α.

stronger model both for discrete and continuous-output cases.

C.5 Subword Embeddings for CoNMT

We rely on the unigram language model for subword segmentation (Kudo, 2018) to train discrete and continuous-output NMT models as mentioned in Section 5. We hypothesize that it is harder for the continuous-output model to predict subwords than for the discrete model. Table 5 illustrates that the f1 macro average for the beginning of the spm tokens and continuation of the spm tokens differ a lot for discrete and continuous models. While the discrete model performs better on continuations,

model	`ro-en`	`en-tr`
discrete words	28.5	8.9
continuous random words	27.6	5.6
discrete tokens	32.1	12.7
continuous random tokens	29.2	9.3

Table 4: BLEU scores for word level and tokens level models on validation set with greedy decoding.

continuous models struggle with continuations of subwords. However, overall scores for pre-trained and random targets are the same for continuation and random embeddings performs slightly better on the beginning of the subwords.

model	F1	
	SPM start	SPM cont.
discrete	0.12	0.14
pre-trained embeddings	0.10	0.09
random embeddings	0.11	0.09

Table 5: F1 score on `newstest2016` `ro-en` for beginning and continuation of the SentencePiece tokens.

D Beam Search

Implementing beam search meaningfully for CoNMT is possible by using the following probabilistic interpretation of the cosine loss as a Langevin: log-likelihood with constant concentration parameter κ: in beam search we use this probabilistic interpretation and take

$$\log p(y_i = t \mid \boldsymbol{y}_{<i}, \boldsymbol{x}) = -\cos(\boldsymbol{E}(t), \boldsymbol{h}) + \log C_d(1),$$

i.e., we apply the normalizing constant of the Langevin distribution for dimension d and fixed concentration $\kappa = 1$. We may then use the built-in `fairseq` beam search using this log-likelihood. We limit the maximum translation length to source length plus 200.

One possible explanation why random embeddings perform better than pre-trained, especially for beam sizes greater than one, may be related to disentanglement: If the continuous output prediction is "off-target" by enough to cause the nearest embedding to be wrong, provided sufficient separation between embeddings, expanding the search to more nearest neighbors can recover the solution. In contrast, for clumped pre-trained embeddings, many embeddings concentrate close to the correct one, polluting the beam.