On Target Representation in Continuous-output Neural Machine Translation

Evgeniia Tokarchuk
Language Technology Lab
University of Amsterdam
e.tokarchuk@uva.nl

Vlad Niculae
Language Technology Lab
University of Amsterdam
v.niculae@uva.nl

Abstract

Continuous generative models proved their usefulness in high-dimensional data, such as image and audio generation. However, continuous models for text generation have received limited attention from the community. In this work, we study continuous text generation using Transformers for neural machine translation (NMT). We argue that the choice of embeddings is crucial for such models, so we aim to focus on one particular aspect: target representation via embeddings. We explore pretrained embeddings and also introduce knowledge transfer from the discrete Transformer model using embeddings in Euclidean and non-Euclidean spaces. Our results on the WMT Romanian-English and English-Turkish benchmarks show such transfer leads to the best-performing continuous model.

1 Introduction & Related work

Discrete neural models represent the majority of systems used in sequence-to-sequence tasks (Sutskever et al., 2014; Vaswani et al., 2017). Despite the promising advantages of continuous-output models in terms of efficiency and expressivity, literature has awarded them relatively little attention. While past work focuses on continuous training objectives, we remark that the choice of word representations is essential.

Continuous-output NMT was first studied by Kumar and Tsvetkov (2019). They study regularized probabilistic loss functions, even though their results show that by far the biggest gain comes from switching to pretrained fastText (Bojanowski et al., 2017) embeddings from word2vec (Mikolov et al., 2013). Bhat et al. (2019) follow up with a study of margin-based losses. However, to the best of our knowledge, there is no comprehensive study on token-level representation and their impact on the continuous NMT performance.

In our work, we attempt to fill the gap and give insights about target representation in continuous-output NMT by highlighting an analogy between target representations and the output layer of a discrete model. We propose, as a knowledge transfer strategy, pretraining word representations with a discrete translation model. On two different language pairs, namely Romanian-English (Ro→En) and English-Turkish (En→Tr), we find that this strategy outperforms externally-trained representations, even from massive pretrained language models. Moreover, we find, somewhat surprisingly, that high dimensionality not only does not help, but can even substantially hurt, and that taking into account the natural spherical geometry of the cosine objective can lead to better performance with smaller dimensionality.

2 Continuous-output NMT

NMT seeks to translate a sequence of tokens $\boldsymbol{x}_{1:N} = (x_1, ..., x_N)$ from the source language to a sequence $\boldsymbol{y}_{1:T} = (y_1, ..., y_T)$ in the target language using a neural model:

$$\boldsymbol{x}_{1:N} \rightarrow \boldsymbol{y}_{1:T}(\boldsymbol{x}_{1:N}) = \arg\max_{\boldsymbol{y}_{1:T}} p(\boldsymbol{y}_{1:T} | \boldsymbol{x}_{1:N}). \quad (1)$$

The probabilistic model above is typically implemented by sequence-to-sequence deep neural models (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017), using the decomposition

$$p(\boldsymbol{y}_{1:T} | \boldsymbol{x}_{1:N}) = \prod_{i=1}^{T} p(y_i \mid \boldsymbol{y}_{1:i-1}, \boldsymbol{x}_{1:N}). \quad (2)$$

In a **discrete model**, the conditional token probabilities in eq. (2) are categorical distributions over a fixed vocabulary $\mathcal{V}_{\text{tgt}}$,

$$\begin{aligned} p(y_i \mid \boldsymbol{y}_{1:i-1}, \boldsymbol{x}_{1:N}) &= \frac{\exp \boldsymbol{e}_{y_i}^\top \boldsymbol{W} \boldsymbol{h}_i}{\sum_{j=1}^{|\mathcal{V}_{\text{tgt}}|} \exp \boldsymbol{e}_{y_j}^\top \boldsymbol{W} \boldsymbol{h}_i} \\ &= \frac{\exp \boldsymbol{h}_i \cdot \boldsymbol{w}(y_i)}{\sum_{v \in \mathcal{V}_{\text{tgt}}} \exp \boldsymbol{h}_i \cdot \boldsymbol{w}(v)}, \end{aligned} \quad (3)$$

where $\boldsymbol{h}_i \in \mathbb{R}^d$ is the model output for position i, (a function of $\boldsymbol{x}$ and the Transformer weights θ), and

Proceedings of the 7th Workshop on Representation Learning for NLP, pages 227 - 235

$\boldsymbol{w}(v) \in \mathcal{W}$ is the embedding of vocabulary token v, *i.e.*, the vth row of $\boldsymbol{W}$. Typically, $\mathcal{W} = \mathbb{R}^d$ and $\boldsymbol{W}$ is randomly initialized and learned jointly with θ. The log-probability of the gold token is typically referred as the *cross-entropy loss*, and has the value:

$$L_{\mathrm{D}}(\theta, \boldsymbol{W}) = -\sum_{i=1}^{T} \log p(y_i \mid \boldsymbol{y}_{1:i-1}, \boldsymbol{x}_{1:N}) = \sum_{i=1}^{T} \left(-\boldsymbol{h}_i \cdot \boldsymbol{w}(y_i) + \log \sum_{v \in \mathcal{V}_{\mathrm{tgt}}} \exp \boldsymbol{h}_i \cdot \boldsymbol{w}(v) \right).$$

In a **continuous model**, the output space is not limited to a discrete vocabulary but instead gives mass to the entire space $\mathcal{W}$, and we interpret the notation $p(y_i|\boldsymbol{y}_{1:i-1}, \boldsymbol{x})$ to mean $p(\boldsymbol{w}(y_i)|\boldsymbol{y}_{1:i-1}, \boldsymbol{x})$. A common parametrization uses the cosine similarity,

$$p(\boldsymbol{w}(y_i)|\boldsymbol{y}_{1:i-1}, \boldsymbol{x}) \propto \exp \frac{\boldsymbol{h}_i \cdot \boldsymbol{w}(y_i)}{\|\boldsymbol{h}_i\| \|\boldsymbol{w}(y_i)\|}. \quad (4)$$

Here, the distribution is over a continuous space, so the normalizer is an integral $\int_{\mathcal{W}} \mathrm{d}\boldsymbol{v} \exp \frac{\boldsymbol{h}_i \cdot \boldsymbol{v}}{\|\boldsymbol{h}_i\| \|\boldsymbol{v}\|}$. By a symmetry argument, it can be shown that the normalizer does not depend on $\boldsymbol{h}$ and is therefore a constant, yielding the **cosine distance loss**:

$$L_{\mathrm{C}}(\theta) = -\sum_{i=1}^{T} \log p(\boldsymbol{w}(y_i)|\boldsymbol{y}_{1:i-1}, \boldsymbol{x}) = \mathrm{const} + \sum_{i=1}^{T} \left(1 - \frac{\boldsymbol{h}_i \cdot \boldsymbol{w}(y_i)}{\|\boldsymbol{h}_i\| \|\boldsymbol{w}(y_i)\|} \right). \quad (5)$$

The cosine loss is an intuitive choice with a history of use in NLP (Subramanian et al., 2018; Wieting et al., 2019). Its probabilistic interpretation we give has roots in directional statistics (Mardia et al., 2000), and corresponds to a Langevin distribution (also known as vMF) with fixed scale. Kumar and Tsvetkov (2019) studied more general Langevin distributions for NMT. Even though these more flexible formulations provide useful modelling extensions, the impact of the loss seems less than the impact of embeddings.

Unlike the discrete model, where the embeddings $\boldsymbol{w}(\cdot)$ can be learned from scratch, in a continuous model, this is not an option because the trivial solution of setting them all to the same (nonzero) value and learning to always output that value as $\boldsymbol{h}_i$ leads to the minimal loss of zero. *Therefore, for continuous-output NMT, good pretrained token representations are essential!*

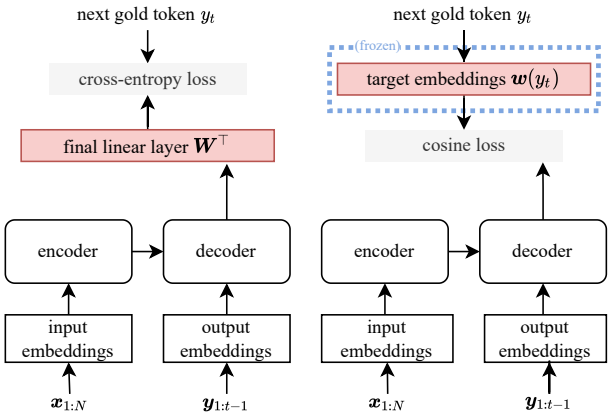

Figure 1: Illustration of the parallels between the discrete (left) and continuous (right) Transformers.

Model architecture. We build our continuous model on top of the Transformer (Vaswani et al., 2017) encoder-decoder model, which powers most state-of-the-art NMT models. In contrast, previous work uses recurrent models (Bahdanau et al., 2015). The encoder is unchanged, while the decoder is slightly reorganized, as shown in figure 1. We re-interpret the output layer $\boldsymbol{W}$ as the target embeddings, which only needs to be applied to the gold token during training. The target embeddings are frozen and set to one of the choices discussed in §3.

3 Target Embeddings

3.1 Euclidean Representations

fastText. Following Kumar and Tsvetkov (2019) we use `fastText` (Bojanowski et al., 2017) target embeddings. We experiment with two different variants. The first is the publicly-available CommonCrawl pretrained `fastText` model (Mikolov et al., 2018; Grave et al., 2018). These models contain subword information and we use the provided API to extract vectors for every subword in the preprocessed MT training data. For comparison, we also train `fastText` models entirely from scratch on the preprocessed MT training data.

mBART. Since the work of Kumar and Tsvetkov (2019), large language models proved highly effective at generating contextualized vector representations for a variety of downstream tasks. We therefore consider extracting target representations from mBART (Tang et al., 2021). For further adaptation to MT, we use the fine-tuned NMT many-to-many mBART-large many-to-many model (Tang et al., 2021) from the huggingface Transformers library (Wolf et al., 2020). A natural thought would be to extract the mBART input

embeddings for subwords occuring in the MT data. However, we found that mBART input embeddings are less adequate that mBART model outputs, especially for subwords that are common in multiple languages, and lead to the poor performance. We refer to the appendix D for details. Therefore, we propose encoding every subword type $v \in V$ by processing `[target-lang] v` through the mBART decoder, and using the last hidden activations.

MT-transfer. Using our observation of the parallel between the linear output layer of a discrete MT model $\boldsymbol{W}$ and the target embeddings in a continuous one (figure 1), we propose a novel knowledge transfer strategy. We train a Transformer-base model (baseline) on the preprocessed MT parallel data, choose the best checkpoint on development set, and use the output layer weights as target embeddings.

3.2 Non-euclidean Representations

Both embedding methods discussed so far assume that the tokens live in an Euclidean space, like most NLP models. However, this assumption is receiving increasing scrutiny (Nickel and Kiela, 2017; Bronstein et al., 2017; Tifrea et al., 2019). Indeed, since the cosine distance is a function of *directions* only, it may be suboptimal to use embeddings that encode information in vector lengths. We consider two methods for learning embeddings on the surface of the sphere, $\boldsymbol{w}(y) \in \mathbb{S}^{d-1} \subseteq \mathbb{R}^d$, where

$$\mathbb{S}^{d-1} := \{\boldsymbol{u} \in \mathbb{R}^d : \|\boldsymbol{u}\| = 1\}. \tag{6}$$

Spherical Text Embeddings (JoSe). Meng et al. (2019) propose learning directional embeddings on the unit sphere using Riemannian optimization, reporting improved performance on word similarity tasks, where cosine similarity is typical. Since continuous MT models also rely on cosine similarity, we expect similar results. We train spherical embeddings using the code released by Meng et al. (2019) on the target-side monolingual data of each MT language pair, after BPE tokenization. The released pretrained JoSe model does not apply, due to lack of subword information.

Spherical MT embeddings. As a spherical counterpart of the MT transfer learning insight, we propose training a baseline Transformer model with decoder input and output embeddings constrained to $\mathbb{S}^{d-1}$. We employ Riemannian optimization (Gabay, 1982; Udriste, 1994; Bonnabel, 2013); specifically, Riemannian Adam (Becigneul and Ganea, 2019) for the last hidden layer $\boldsymbol{W}$ as well as the other embeddings, and regular (Euclidean) Adam (Kingma and Ba, 2015) for all other parameters. Riemannian Adam is provided in `geoopt` (Kochurov et al., 2020). To our knowledge, this is the first instance of non-euclidean embeddings trained with an MT objective.

3.3 Dimensionality Reduction

While high-dimensional vectors can be richer, computational costs increase with dimension, and distances can be harder to tell apart (Aggarwal et al., 2001; Beyer et al., 1999).

To explore the impact of the target dimension, for the embeddings trained only on MT data, we retrain the embeddings for every dimensionality we consider. For external embeddings, we use PCA: in the case of `fastText`, we use the provided `reduce_model.py` script. For mBART, we apply cosine kernel PCA (Schölkopf et al., 1997) from `scikit-learn` (Pedregosa et al., 2011). Dimensionality reduction on the sphere is non-trivial and a possible avenue for future work.

4 Experiments

We experiment using the publicly available WMT 2016 Ro→En dataset with 612K parallel training sentences, and the WMT 2018 En→Tr dataset with 207K parallel training sentences. We compute BLEU (Papineni et al., 2002) using `sacrebleu` (Post, 2018)[1] on `newsdev2016` and `newstest2016` for both Ro→En and En→Tr. Detailed information about data is collected in appendix A.

All experiments and implementation are based on fairseq (Ott et al., 2019) framework. We use 6-layers Transformer base model as a baseline. For continuous model, encoder and decoder embeddings size are set to 512 (they are not initialized with pretrained embeddings), and output layer size depends on the target embeddings dimensionality. We choose the best model checkpoint based on development BLEU. For generation, we rely on the top-1 nearest neighbor search (greedy) using cosine similarity, the details are discussed in appendix C.

4.1 Results & Analysis

Table 1 shows the BLEU along with the BERTScore (Zhang et al., 2020) results of continuous output NMT models with different target embeddings. Since BERTScore is based on semantic similarity, it is suitable to assess the continuous model

[1] BLEU+case.mixed+numrefs.1+smooth.exptok.13a+version.1.5.1

embeddings	dim.	Ro→En dev16 BLEU	Ro→En dev16 BSc	Ro→En test16 BLEU	Ro→En test16 BSc	En→Tr dev16 BLEU	En→Tr dev16 BSc	En→Tr test16 BLEU	En→Tr test16 BSc	En→Tr test17 BLEU	En→Tr test17 BSc
discrete	-	33.0	65.6	31.6	64.9	12.0	69.3	12.2	69.2	12.2	69.8
+beam=5	-	33.7	66.6	32.3	66.1	12.7	70.4	12.8	70.5	13.0	71.0
Trained on target monolingual data											
JoSe (𝕊)	100	29.6	43.3	27.4	43.1	2.7	54.1	2.9	54.7	3.3	55.9
JoSe (𝕊)	50	29.9	50.9	28.2	51.8	9.7	64.0	9.4	63.9	9.9	64.7
fastText	512	26.4	47.0	25.4	47.9	3.5	52.8	3.3	54.1	3.3	52.7
fastText	300	27.2	51.4	26.6	52.1	9.1	64.0	9.0	63.9	9.5	64.7
fastText	100	29.3	57.1	28.6	57.2	9.2	62.6	9.2	62.6	9.4	63.1
fastText	50	29.3	56.4	28.6	56.5	9.2	63.1	9.2	63.1	9.4	63.8
Trained on bilingual data											
MT-transfer	512	29.7	56.4	28.7	57.2	10.9	67.9	10.7	67.8	11.3	68.6
MT-transfer	100	**32.2**	**63.0**	**30.9**	**62.9**	8.5	61.8	8.2	61.5	8.9	62.3
MT-transfer	50	31.7	62.3	30.6	62.3	8.5	60.8	8.6	60.7	8.9	61.4
MT-transfer (𝕊)	512	30.4	61.0	29.0	60.9	10.3	67.1	9.8	66.8	10.2	67.6
MT-transfer (𝕊)	100	30.8	61.0	29.7	60.9	**11.4**	**68.6**	**11.2**	**68.1**	**11.6**	**69.1**
MT-transfer (𝕊)	50	31.3	60.9	30.0	60.9	9.2	63.3	9.1	62.8	9.5	63.5
Pretrained on external data											
fastText	300	27.5	55.1	27.0	55.7	9.2	62.6	9.1	62.1	9.3	63.0
$\text{fastText}_{\text{PCA}}$	100	29.6	59.4	28.6	59.0	9.1	63.0	9.3	62.8	9.5	63.5
mBART-MT	1024	24.9	48.6	24.6	49.5	0.0	29.5	0.0	29.6	0.0	29.5
$\text{mBART-MT}_{\text{PCA}}$	512	29.5	58.9	28.7	59.5	9.5	65.6	8.9	64.5	9.2	65.2
$\text{mBART-MT}_{\text{PCA}}$	100	28.9	57.1	27.9	58.0	9.7	65.1	9.2	64.5	9.8	65.3
$\text{mBART-MT}_{\text{PCA}}$	50	27.3	54.2	26.4	54.1	8.2	61.8	7.9	61.4	8.5	62.2

Table 1: BLEU and BERTscore (BSc), in percentages, on newstest and newsdev. Spherical models are denoted by 𝕊.

performance. We re-scale BERTScore using baseline, to provide more human-readable outputs. The BERTScores agrees with the BLEU score both on Ro→En and En→Tr. Contrary to past work (Kumar and Tsvetkov, 2019; Bhat et al., 2019), when upgrading to state-of-the-art Transformer models with BPE, continuous models do not catch up to the discrete counterpart. We attribute this to the highly tuned Transformer architecture, and find that our exploration manages to shrink the gap considerably. We next analyze the various dimensions of variation in the choice of target representation.

MT knowledge transfer. On both tasks, the best performing continuous model uses embeddings learned by a discrete MT model. Bilingual data contains valuable information about the target language, but external mBART embeddings lag behind MT-transfer, perhaps since the latter are fine-tuned to the target language and domain. This finding prompts promising directions for hybrid embeddings via fine-tuning or adaptation.

Geometry. Spherical embeddings (JoSe and MT-transfer(𝕊)) prove useful compared to the euclidean embeddings, and tend to scale well to smaller dimensions and datasets. MT-transfer(𝕊) is the best continuous model for En→Tr.

Dimensionality. Throughout, we record the best performance with embeddings slightly smaller than the standard values used in discrete models. This is most pronounced for mBART-MT, with which En→Tr training fails entirely for $d = 1024$. According to our findings, the smaller dimensionality of the target embeddings benefits the model's performance. However, it might no longer hold for large-scale MT datasets.

External pretraining. Surprisingly, we find no clear indication that large-scale external pretraining with fastText or mBART is superior to leaning only on the task data, even when compared to monolingual embeddings, and even on the lower-resource language pair. However, we cannot use the full contextualization abilities of mBART, because we are limited to selecting one embedding

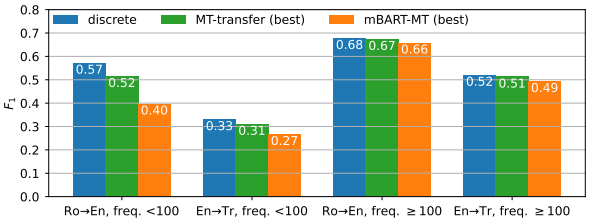

Figure 2: Word-level F_1 score by training frequency.

	Output
Src.	În Bucuresti se vor inregistra 26 de grade la amiaza.
Ref.	Bucharest will register 26 degrees at noon.
discrete	Bucharest will register 26 degrees at noon.
MT-transfer	There will be 26 degrees at afternoon in Bucharest.
mBART-MT	There will be 26 degrees in Bucharest at evening.
Src.	horă și rock cu vioară și chitară
Ref.	Hora and rock with a violin and guitar
discrete	Hora and rock with both a violin and guitar
MT-transfer	Hora, rock with vivid and chitar
mBART-MT	Resolution and rock with its shadow and furniture

Table 2: Translation examples. Words with training frequency < 100 are highlighted.

vector per target subword. Better transfer of contextual representations from large language models remains an open question.

Rare words. One might expect external pretraining to benefit words that occur rarely in the MT training data, via transfer. Figure 2 reveals the opposite trend. Even the best continuous model struggles for words with frequency under 100, but mBART-MT degrades much more for such rare words. For more common words, the gap is small. Some examples of sentences with the rare words are shown in Table 2. More examples can be found in Appendix E.

Length. We find continuous models to struggle more with shorter sentences. For Turkish target sentences longer than 10 words, the difference in average sentence BLEU between the discrete and the best continuous model is 1.04; for sentences with ≤10 words it is 2.48. Ro→En exhibits a similar trend. This suggests future work should focus on the representations of rare words and short sentences.

5 Conclusion

In this work, we investigated the importance of target representations for continuous NMT in two language pairs. We find that our proposed strategy to transfer embeddings from a discrete Transformer model outperforms all other embedding choices. We pinpoint the impact of properties like dimensionality and geometry, and provide further insight into the errors made by continuous models. Our proposed transfer strategy is effective despite using much less data compared to large pretrained models. We believe that further research into combining external data with MT-transfer embeddings may be necessary for improving continuous model performance. Even though our model performance is behind the discrete model, we argue that this work can be seen as a stepping stone for building strong and reliable continuous model for text generation.

Acknowledgments

We thank all the members of the UvA Language Technology Lab for their constant feedback on our work. Special thanks to Ali Araabi and Amir Soleimani for their useful comments on the manuscript, and to Nicola De Cao for the insightful discussions on the topic. Finally, we want to thank anonymous reviewers for their valuable input and suggestions. Vlad Niculae is partially supported by the Hybrid Intelligence Centre, a 10-year program funded by the Dutch Ministry of Education, Culture, and Science through the Netherlands Organisation for Scientific Research (https://hybrid-intelligence-centre.nl).

References

Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. 2001. On the surprising behavior of distance metrics in high dimensional space. In *Proceedings of the International Conference on Database Theory*. Springer.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Gary Becigneul and Octavian-Eugen Ganea. 2019. Riemannian adaptive optimization methods. In *International Conference on Learning Representations*.

Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1999. When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer.

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.

Silvere Bonnabel. 2013. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229.

Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

Daniel Gabay. 1982. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37(2):177–219.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Max Kochurov, Rasul Karimov, and Serge Kozlukov. 2020. Geoopt: Riemannian optimization in PyTorch.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

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

Kanti V Mardia, Peter E Jupp, and KV Mardia. 2000. *Directional statistics*, volume 2. Wiley Online Library.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. 2019. Spherical text embedding. *Advances in Neural Information Processing Systems*, 32.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*.

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

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, volume 30.

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.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. 1997. Kernel principal component analysis. In *Proceedings of the International Conference on Artificial Neural Networks*, Berlin, Heidelberg. Springer Berlin Heidelberg.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. In *Proceedings of the International Conference on Learning Representations*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, Cambridge, MA, USA. MIT Press.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. 2019. Poincare Glove: Hyperbolic word embeddings. In *Proceedings of the International Conference on Learning Representations*.

Constantin Udriste. 1994. *Convex Functions and Optimization Methods on Riemannian Manifolds*, volume 297. Springer Science & Business Media.

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.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond BLEU:training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

A Data

We follow a standard pre-processing pipeline: all training sentences are tokenized and truecased using moses. We apply BPE (Sennrich et al., 2016) segmentation with 40K merge operations for Ro→En and 16K for En→Tr. Where necessary, we apply the SPM (Kudo and Richardson, 2018) model provided by the mBART pretrained model. The training data statistics are collected in Table 3.

Validation set `newsdev2016` and test set `newstest2016` for Ro→En contains 1999 sentences. Validation set `newsdev2016` and test set `newstest2016` for En→Tr contains 1999 sentences. En→Tr validation set `newsdev2016` contains 1001 sentences, test set `newstest2016` contains 3000 sentences and `newstest2017` contains 3007 sentences.

	Ro→En	En→Tr
# train sentences	612K	207k
# running tokens (tgt)	16.6M	4.6M
target vocab. size	25k	12K

Table 3: Training data statistics

B Hyperparameters

For all models, the learning rate is set to $5 \cdot 10^{-4}$ and the effective batch size set to 64k tokens. Warm-up steps are 10K for Ro→En and 4k for En→Tr. We use dropout 0.3 for all our models. We train model with the Adam optimizer (Kingma and Ba, 2015).

C Generation

To find the closest token on each generation step, we use the cosine similarity between output of the model and target embeddings.

$$\tilde{y}_i = \underset{v \in V_{tgt}}{\arg\min} d(\boldsymbol{h}_i, \boldsymbol{w}(v)) \tag{7}$$

where $\tilde{y}_t$ is the token predicted by the model, and $d(\cdot)$ is the cosine distance between the model output and the token embeddings of the token in target vocabulary.

The complexity of the NN search for NMT depends on vocabulary size, the sequence length and the vector dimensions. To speed up search, we use the `faiss` (Johnson et al., 2019) library for fast nearest neighbors search. However, instead of approximation, we use exact search, which nevertheless boosts the computation speed. Investigation of the different variants of the approximate nearest neighbors search is out of the scope of this paper.

D mBART embeddings

As we mentioned in §3.1, the straightforward way to utilize the mBART embeddings is to extract the input embeddings matrix. The extracted embeddings matrix contains 250K vocabulary types. We filter embeddings to keep only the tokens, which is observed in training MT data. After filtering, the vocabulary consists of 27,508 types. However, the performance of continuous models using these embeddings drop dramatically on Ro→En (17.0 BLEU on the development set, which is 16.7 BLEU worse than a discrete model). We hypothesize that this might be due to the multilingual ambiguity of the token embeddings in the input matrix. For the filtered embeddings matrix, the 3 nearest neighbors for the word "_neighbor" are: "_neighborhood", "_mondat", "_mbr". For mBART-MT, obtained as discussed in §3.1, the 3 nearest neighbors for the word "_neighbor" are: "friend", "_companion" and "_mentor".

E Examples

We provide sentence examples of the best performing model for each embeddings type in table 4 on the next page.

	Output
Src.	În Bucuresti se vor inregistra 26 de grade la amiaza.
Ref.	Bucharest will register 26 degrees at noon.
discrete	Bucharest will register 26 degrees at noon.
JoSe (𝕊)	There will be 26 degrees at afternoon in Bucharest.
fastText	There will be 26 degrees in Bucharest at afternoon.
MT-transfer	There will be 26 degrees at afternoon in Bucharest.
MT-transfer(𝕊)	There will be 26 degrees in Bucharest at the afternoon.
fastText (pretrained)	There will be 27 degrees in Bucharest in the afternoon.
mBART-MT	There will be 26 degrees in Bucharest at evening.
Src.	The other undergraduates giggled.
Ref.	Diğer lisans öğrencileri kıkırdadı.
discrete	Diğer lisans öğrencileri de oldukça yavaş gitti.
JoSe (𝕊)	Diğer başka leme eğitim ları da zevkler.
fastText	Diğer mezunlar da karmaşıklaştırıldı.
MT-transfer	Diğer mezunlar ise hediye ediliyorlar.
MT-transfer (𝕊)	Diğer mezunlar ise bıkmış durumda.
fastText (pretrained)	Diğer mezunlar ise relayor.
mBART-MT	Diğer lisans öğrencileri beenhard.

Table 4: Translation examples for Ro→En and En→Tr. Continuous models have a tendency to select synonyms or near-synonyms (noon and afternoon, öğrencileri and mezunlar.)