Incremental Vocabularies in Machine Translation Through Aligned Embedding Projections

Salvador Carrión(✉) and Francisco Casacuberta(✉)

Universitat Politècnica de València, Camí de Vera, 46022 València, Spain
{salcarpo,fcn}@prhlt.upv.es

Abstract. The vocabulary of a neural machine translation (NMT) model is often one of its most critical components since it defines the numerical inputs that the model will receive. Because of this, replacing or modifying a model's vocabulary usually involves re-training the model to adjust its weights to the new embeddings.

In this work, we study the properties that pre-trained embeddings must have in order to use them to extend the vocabulary of pre-trained NMT models in a zero-shot fashion.

Our work shows that extending vocabularies for pre-trained NMT models to perform zero-shot translation is possible, but this requires the use of aligned, high-quality embeddings adapted to the model's domain.

Keywords: Neural machine translation · Zero-shot translation · Continual learning

1 Introduction

Machine translation is an open vocabulary problem. There are many approaches to this problem, such as using vocabularies of characters, subwords, or even bytes.

Even though these approaches solve many of the open-vocabulary problems, rare or unknown *words* are still problematic because there is little to no information about them in the dataset. As a result, the model cannot learn good representations about them. For example, if we create a character-based vocabulary of lower and upper case letters but our training set only contains lower case letters, our model will not be able to learn any information about uppercase letters despite being in its vocabulary. Similarly, if they appear just a couple of times, the model will also not be able to learn enough information about how to use them.

As there are always constraints when training a model (e.g., amount of data or computational resources), we need a method to update these vocabularies after our model has been trained, but without the need to re-train the whole

Supported by Pattern Recognition and Human Language Technology Center (PRHLT).

© Springer Nature Switzerland AG 2022
A. J. Pinho et al. (Eds.): IbPRIA 2022, LNCS 13256, pp. 27–40, 2022.
https://doi.org/10.1007/978-3-031-04881-4_3

model again or without the risk of shifting its domain or weight distribution due to the initialization and re-training of the new entries.

Most approaches to extend a model's vocabulary either replace the old embeddings with new ones (randomly initialized) or add new entries (randomly initialized) on top of the old ones. However, the problem with these approaches is that they require the fine-tuning of the model, thing that is not possible when there is no data, or when there is so little data that its fine-tuning might hurt the performance of the model by shifting its weight distribution too much from the previous one.

In this work, we first study the properties that word embeddings must have in order to use them in a zero-shot scenario. We do this by studying the convergence of different word embeddings from several domains and qualities, projected to a common latent space (e.g., GloVe, FastText and custom ones). Then, we use FastText as a workaround to generate aligned embeddings for unknown words and extend the vocabulary of a model in a zero-shot manner. Finally, we adapt these embeddings to the target domain, achieving a significant boost in performance for our zero-shot models.

The contributions of this work are two-fold:

- First, we show that word embeddings from different domains do not appear to converge regardless of their quality. Hence, this suggests that latent space alignment is a requirement for zero-shot translation.
- Then, we show that it is possible to improve the performance of an NMT model in a zero-shot scenario by extending its vocabulary with aligned embeddings. However, to achieve significant gains in performance, these embeddings must be adapted to the target domain.

2 Related Work

In recent years, many approaches have been proposed to learn good word representations from large corpora. Among them, we can highlight Word2Vec [18], which was one of the first successful neural models to learn high-quality word vectors from vast amounts of data using neural networks; GloVe [21], which achieved state-of-the-art performance by learning through a matrix factorization technique the co-occurrence probability ratio between words as vector differences; and FastText [5], which extends Word2Vec to represent words as the sum of n-gram vectors.

Despite the success of these approaches, static word embeddings could not capture the meaning of a word based on its context. To deal with problem[1], researchers began to focus their efforts on contextualized word embeddings such as ELMo [22], a deep bidirectional language model, or BERT [9], a language model that make use of the encoder part of Transformer architecture [27].

The zero-shot problem in machine translation has been widely studied with relative success. For example, GNMT [12] showed that zero-shot machine translation was possible through multilingual systems; Ha et al. [11] presented strategies

[1] This is typically seen in *homograph* words.

with which to improve multilingual systems to tackle zero-shot scenarios better; and Zhang et al. [31] improved the zero-shot performance by using random online back-translation to enforce the translation of unseen training language pairs.

In contrast, the problem of lifelong learning in neural machine translation has not received enough attention, and it was not until 2020 that it was introduced as a task in the WMT [3] conference. Following this research line, Xu et al. [29] proposed a meta-learning method that exploits knowledge from past domains to generate improved embeddings for a new domain; Liu et al. [15] learned corpus-dependent features by sequentially updating sentence encoders (previously initialized with the help of corpus-independent features) using Boolean operations of conceptor matrices; and Qi et al. [24] showed that pre-trained embeddings could be effective in low-resource scenarios.

3 Models

3.1 Tokenization

Most tasks related to natural language require some form of pre-processing, being tokenization one of the most common techniques. There are many tokenization strategies, such as the ones based on bytes [10,30], characters [8,19,28], subwords [14,25], or words [7,26]. However, given the purpose of our work, we considered that word-based tokenization would be enough.

Specifically, we used the Moses tokenizer [13] because it is a reversible word-based tokenizer. That is, it can transform a tokenized text into its original version, and it also considers punctuation marks and nuances about the language.

3.2 Transformer Architecture

Nowadays, most machine translation systems are based on encoder-decoder neural architectures. However, amongst these Seq-to-Seq architectures [26], the Transformer [27] is the one that has obtained the latest state-of-the-art results.

The Transformer model has the advantage that it is based entirely on the concept of *attention* [4,16] to draw global dependencies between the input and output. Because of this, the Transformer does not need to make use of any recurrent layers to deal with temporal sequences, and it can process its sequences in parallel, obtaining significant performance improvements.

To accomplish this, the Transformer uses linear and normalization layers, residual connections, attention mechanisms, and mask-based tricks to encode the temporal information of its sequences.

3.3 Projecting Vectors into Different Latent Spaces

One of the requirements to extend the vocabulary of a pre-trained model with new entries is that the dimensionality of the new entries matches the dimensionality of the old ones.

If both entries have the same dimensionality, there is no problem, but if they do not, we need to transform them to have the same dimensionality (i.e., vector compression). There are many ways to project a set of vectors into a lower-dimensional latent space. We decided to explore the most common techniques, such as the projection of embeddings through PCA and Autodercoders (linear and non-linear), and their reconstruction error.

It is essential to highlight that although two sets of vectors might have the same dimensionality, their vectors might be projected into different regions of the same latent space. Namely, they could show similar distances amongst their respective word-pairs, but at the same time, they could reside in different locations.

Principal Component Analysis. Principal Component Analysis (PCA) is a method commonly used for dimensionality reduction. This method allows us to project an n-dimensional data point from a given dataset into another vector that resides in another n-dimensional space but with lower dimensionality while preserving as much of the data's variation as possible.

We will use this method to transform the embeddings from their original dimension to a lower dimension to match the one that our models need.

Autoencoders. Autoencoders are a type of Artificial Neural Network used to learn specific representations from unlabeled data. That is, they can encode a vector in a given n-dimensional space into another n-dimensional space while preserving as much of its information as possible.

These models usually are made from two separated blocks: i) the encoder, which learns how to encode the input data into a latent space; ii) and the decoder, which is in charge of transforming the encoded input data into the original input data.

In this work, we will use linear autoencoders (which in theory should be equivalent to PCA) and non-linear autoencoders. This comparison aims to explore potential non-linear representations for the embeddings.

4 Experimental Setup

4.1 Datasets

The data used in this work comes mainly from the WMT tasks [1,2]. In Table 1 we find all datasets used in this work.

Table 1. Datasets partitions

Dataset	Train	Val	Test
Europarl v7 (DE-EN)	2M/100K	5000	5000
Multi30K (DE-EN)	29K	1014	1000

The values in this table indicate the number of sentences.

These datasets contain parallel sentences extracted from different sources and domains:

- **Europarl** contains parallel sentences extracted from the European Parliament website
- **Multi30k** is a small dataset from WMT 2016 multimodal task, also known as Flickr30k

4.2 Training Details

As we were interested in exploring techniques to expand the vocabularies of a pre-trained model, we did not use subword-based tokenizers such as BPE [25] or Unigram [14]. Instead, we pre-processed our datasets using Moses [13] to create a word-based vocabulary (one per language).

For each dataset, we created word-based vocabularies with 250, 500, 1000, 2000, 4000, 8000, and 16000 words each. Then, we encode the Multi30K (de-en), Europarl-100k (de-en), and Europarl-2M (de-en) datasets using these vocabularies.

In addition to this, we use AutoNMT [6] as our sequence modeling toolkit. All the experimentation was done using a small version of the standard Transformer with 4.1 to 25M parameters depending on the vocabulary size (See Table 2). The reason for this was to speed up our research as the focus of this work was not to achieve state-of-the-art performance but to extend the vocabulary of pre-trained models in a zero-shot manner.[2]

Table 2. NMT models hyperparameters

Model	Parameters	Hyperparameters
Transformer small	4.1M (S)—25.0M (L)	3 layers/8 heads/256 dim/512 ffnn

Common hyperparamters: Loss function: CrossEntropy (without label smoothing). Optimizer: Adam. Batch size: 4096 tokens/batch, Clip-norm: 1.0. Maximum epochs: 50–100 epochs with early stopping (patience = 10). Beam search with 5 beams.

In order to make a fair comparison for all the models studied here, we use the same base transformer with similar (or equal) training hyper-parameters as long as the model was able to train correctly.

[2] All models were trained using 2x NVIDIA GP102 (TITAN XP) - 12 GB each.

4.3 Evaluation Metrics

Automatic metrics compute the quality of a model by comparing its output with a reference translation written by a human.

Due to the sensitivity of most metric systems to their hyper-parameters and implementation, we used SacreBLEU [23] which produces shareable, comparable, and reproducible BLEU scores. Similarly, we also evaluated our models with BERTScore [32], but no additional insights were gained from this.

- **BiLingual Evaluation Understudy (BLEU)** [20]: Computes a similarity score between the machine translation and one or several reference translations, based on the n-gram precision and a penalty for short translations.
- **BERTScore** [32]: Using pre-trained contextual embeddings from BERT, it matches words in candidate and reference sentences by cosine similarity

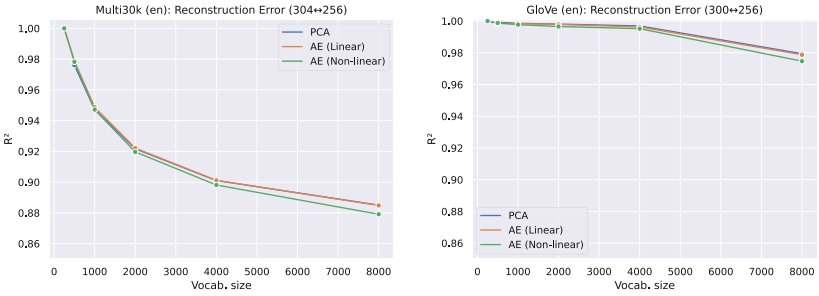
5 Experimentation

5.1 Projecting Pre-trained Embeddings

The typical approach when using pre-trained embeddings is to initialize the model's embedding layers, freeze them, train the model, and finally, fine-tune these layers using a small learning rate. Although effective, we have to consider that the dimensions of the embedding layers of our model and the ones from the pre-trained embeddings must match. This is usually not a problem when training models from scratch as we can leave the original embedding dimension or simply add a linear layer to transform one dimension into another, but for pre-trained models, this could become a non-trivial task.

When adding new embeddings for a pre-trained model, we need to consider a few things. For instance, if we can afford to replace all the embeddings, we can simply drop the old embeddings, add the new ones (plus a linear layer to match dimensions, if needed), and then fine-tune the model. The advantage of this approach is that all vectors share the same latent space. However, if we need to extend an existing embedding layer, not only the dimensions of the old and the new embeddings must match, but the region of the latent space in which their vectors are expected. That is, let say that in a embedding A, the word *dog* is at position $(1, 0)$ and the word *cat* at $(1, 1)$. But in the embedding B, *dog* is at $(-1,\ 0)$ and *cat* at $(-1,\ -1)$. In both cases, their respective distance is 1, and their embedding dimension is 2. However, both embeddings are in different locations of the same latent space. Because of that, it is expected that the new embeddings have to be aligned with the previous ones to use them in a zero-shot environment.

In Fig. 1a we have the reconstruction errors (R^2) for the embeddings of a translation task, corresponding to a Transformer trained on the Multi30k (de-en) datasets. Similarly, in Fig. 1b we have the reconstruction errors (R^2) for the GloVe [21] embeddings, which are high-quality embeddings from modeling co-occurrences probabilities through a matrix factorization process.



(a) Multi30K (304→256) (b) GloVe (300→256)

Fig. 1. Reconstruction errors of the compressed embeddings using Linear and Non-linear methods. Comparing both images, we can see that the higher the quality of the embeddings, the lower the reconstruction error. Furthermore, non-linear methods do not seem to be needed, as the embeddings seem to basically hold linear relations amongst them

In both figures, we see that PCA and autoencoders performed remarkably similarly. For the linear projections, PCA had a perfect overlap with the linear autoencoder, as expected, but the former significantly faster[3]. In contrast, the non-linear autoencoder performed slightly worse than these two, probably due to its additional complexity. From these observations, it seems that there is no need for non-linear methods when it comes to projecting embeddings to other latent spaces due to the linear relations of their vectors.

Concerning the vocabulary size (number of embeddings), as we increase it, the reconstruction error also increases regardless of the model. We believe that this is because the entries in our vocabulary are sorted by frequency, forming a long-tail distribution. Hence, as we increase the vocabulary size, we add less and less frequent entries to our vocabulary. As a result, their embedding vectors will not have the same average quality as the previous ones due to the lack of samples (frequency), so they will be noisy. Because of this, and given that noise cannot be compressed, it is to be expected that some of their information is lost, given that both PCA and Autoencoders perform lossy compression.

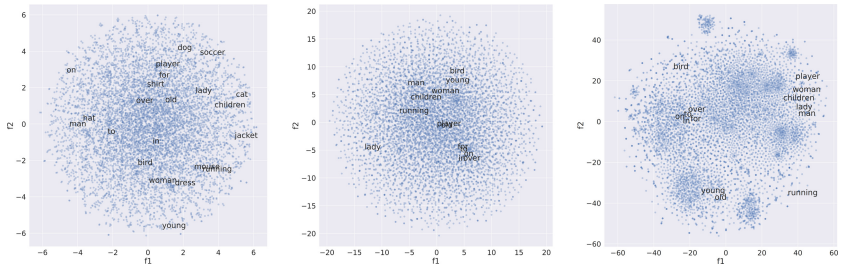
In the case of the GloVe embeddings, we see that the reconstruction errors remained pretty low regardless of the vocabulary size. In the same line as the previous explanation, we think that this is because the GloVe embeddings have more quality than the Multi30k embeddings because the GloVe embeddings resulted from a technique specifically designed to learn high-quality embeddings, while the Multi30K embeddings were the byproduct of a translation model trained on a very small dataset.

[3] We compare these linear methods despite their theoretical equivalence as a sanity check, and because we cannot use PCA when the dimensionality of the vectors is greater than the number of samples.

5.2 On the Importance of High-Quality Embeddings

In order to perform zero-shot translation when new vocabulary entries are added, we need these entries to share the same latent space at some point as an initial requirement.

We expected that as we increased the amount of training data, the embeddings would not only improve their quality but, at some point, they would start to converge to the same regions in the latent space. To test this theory, we projected embeddings from multiple dataset sizes and domains into a 2-dimensional space using t-SNE [17] to visualize the shape and the linear relations between the clusters of words, in order to intuitively get a grasp on both the quality and similarity of the embeddings.



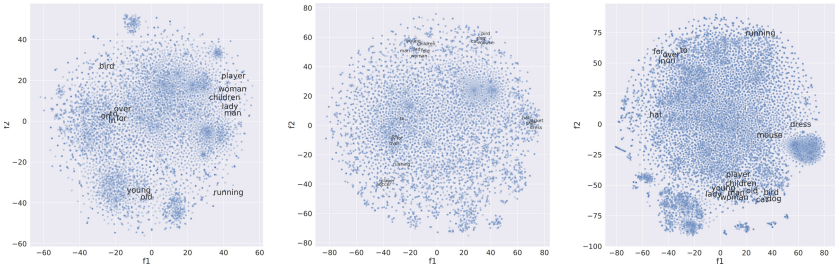
(a) Multi30K (b) Europarl-100k (c) Europarl-2M

Fig. 2. t-SNE projections for the 8000-word embedding layer (en) of a transformer trained on Multi30k, Europarl-100k and Europarl-2M.

In Fig. 2 we see that it was not until we used a dataset with 2M sentences (See Fig. 3a) that we started to see clusters of similar words. However, when we re-trained the model, we ended up with a completely different set of high-quality embeddings (different clusters) that would not have allowed zero-shot translation.

Finally, we wanted further explore the non-convergence problem by adding two well-known high quality embeddings: GloVe [21] and FastText [5]. From the t-SNE projections [17] in Fig. 3 we can see that although these high-quality embeddings group similar words together (i.e., *man, woman, children*), their clusters are completely different as well as the distance between non-related words (i.e., distance between *bird* and *man*).

From these results, it is clear to us that in order to perform zero-shot translation, the new embeddings have to be perfectly aligned with the previous ones. It is essential to highlight that we ended up with different projections each time we trained a t-SNE projection (figures). However, these projections were remarkably similar for the same embedding set (e.g., GloVe-GloVe, FastText-FastText, etc.), but very different between different embedding sets (e.g., GloVe-FastText,



(a) Europarl-2M (b) GloVe (c) FastText

Fig. 3. t-SNE projections for the Europarl, Glove and FastText embeddings

Europarl-FastText). Because of this, we consider that our findings are sound and not the artifact of a specific projection.

5.3 Zero-shot Translation

To study whether it was a strong requirement that the new embeddings had to be aligned with previous ones in order to perform zero-shot translation, we designed three experiments in which we studied vocabulary expansion using:

- Unaligned Embeddings
- Aligned Embeddings (out-of-domain)
- Aligned Embeddings (in-domain)

For each experiment, we trained six translation models with vocabularies of 250, 500, 1000, 2000, 4000, and 8000 words. We then extended these vocabularies to 16K words using pre-trained embeddings, but without fine-tuning (zero-shot). Finally, we evaluated the performance of each model as a zero-shot task before and after the vocabulary expansion.

Unaligned Embeddings. In this experiment, we trained six translation models (with randomly initialized embeddings) on the Europarl-100k (de-en) dataset. Once the models were trained, we extended their vocabularies up to 16K (without fine-tuning) using the following pre-trained embeddings:

- Multi30K (Low-quality/out-of-domain)
- Europarl-100k (Low-quality/in-domain)
- GloVe (High-quality/out-of-domain)

However, the results in all cases were worse than the base model, although not by much, with a decrease of 0.25 to 1.0 pts of BLEU. This happened regardless of the quality of the embeddings (GloVe vs. Multi30k/Europarl-100k) and their domain (Europarl-100k/in-domain vs. GloVe/out-of-domain). Similarly, we

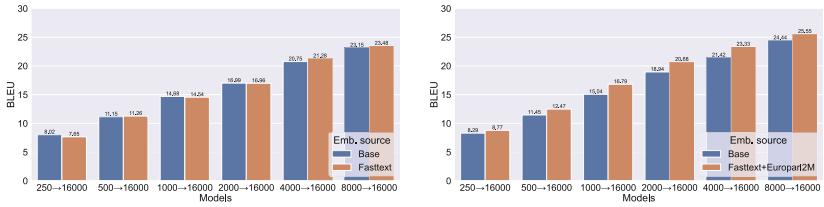
performed another experiment where the new entries had been randomly initialized, but the results were surprisingly similar. This observation shows that an NMT model does not know what to do with the new embeddings without proper alignment or re-training despite their quality.

From these results, we conclude that for an NMT model, the quality and the domain adaptation of these pre-trained embeddings are properties that might be critical for transfer learning but not sufficient to perform zero-shot translation.

Aligned Embeddings. Since in the previous experiment we had shown that the quality and domain adaptation of the embeddings were not a sufficient requirement for their use in zero-shot settings, this time we wanted to study whether, as we previously hypothesized, the alignment of the embeddings was a critical requirement for extending the vocabularies of an NMT model in a zero-shot setting.

To this end, we designed a twofold experiment in which we studied the use of aligned embeddings, but with in-domain and out-of-domain embeddings.

For the experiment with the out-of-domain aligned embeddings, we decided to use FastText to initialize the embeddings of our models[4]. After initializing the embeddings layers and freezing them, we trained six NMT models with vocabularies from 250 to 8K words on the Europarl-100k (de-en) dataset, similar to previous experiments. Finally, we extended the vocabularies of these models up to 16K words. As the previous and new embeddings had been generated using FastText (and not modified), both embeddings were already aligned, so there was no need to use any alignment method.



(a) FastText embeddings (raw) (b) FastText embeddings (fine-tuned)

Fig. 4. BLEU scores from using aligned embeddings: FastText (raw) and FastText (fine-tuned). Adapting the aligned embeddings to the model's domain seem to be particularly relevant.

Interestingly, we can see in Figs. 5 and 4a that when we trained our models using the FastText embeddings (frozen), we got a wide range of results depending

[4] The reason why we switched from GloVe to FastText is that the latter had multilingual support and in addition to this, it allowed us to generate embeddings for unknown words.

on the original vocabulary size. When we extended the vocabulary from 250 to 16K, we lost 0.37 pts of BLEU. Then, the increases or decreases in performance were marginal after expanding the 500, 1K, and 2K vocabularies up to 16K words. In contrast, after we extended the 4K and 8K vocabularies to 16K, we obtained small but consistent performance increases of up to +0.53 pts of BLEU.

These results seem to indicate that even though the embeddings were aligned, an NMT model still needs enough words to learn to generalize to other but similar words. However, we wondered how domain adaptation could affect these results, so we repeated the previous experiment, but this time, fine-tuning the FastText embeddings to the Europarl domain.

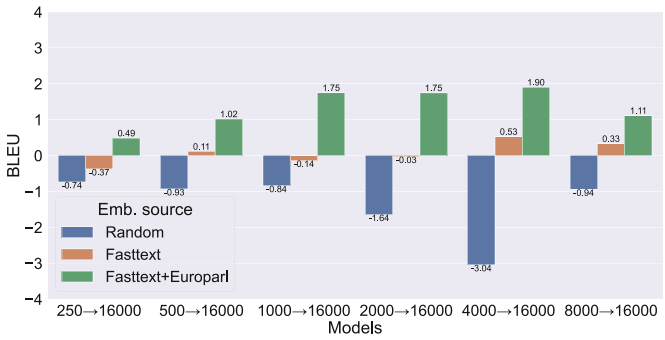


Fig. 5. BLEU scores differences from using Random (baseline) and FastText (raw and fine-tuned) aligned embeddings. Adapting the aligned embeddings to the model's domain seem to be particularly relevant but the model still requires an initial medium vocabulary to learn to generalize to unseen embeddings

Surprisingly, this fine-tuning turned out to be much more important than we initially expected as we got significant improvements in all our models, with a peak improvement of +1.92 pts of BLEU (See Fig. 4b and Fig. 5).

These results appear to indicate that if we want to expand the vocabulary of an NMT model in a zero-shot fashion, both the alignment and domain adaptation of the embeddings are strong requirements for zero-shot translation.

6 Conclusions

In this paper, we have studied the properties required to extend the vocabularies of an NMT model in a zero-shot fashion.

First, we have shown that high-quality embeddings have low-construction errors when projected into lower-dimensional spaces. However, despite the large amounts of data needed to obtain high-quality embeddings, these embeddings do not seem to converge to the same regions of the latent feature space. Hence, embedding alignment is required. Next, we have shown that extending the

vocabulary of an NMT model in a zero-shot fashion is not only possible, but it is also a simple and effective way to improve the performance of an NMT model without re-training.

Finally, our work suggests that to expand the vocabulary of an NMT model and perform zero-shot translation, we must use high-quality embeddings adapted to the domain and, above all, properly aligned with the previous embeddings.

7 Future Work

Many adaptations and experiments have been left for the future due to a lack of time. Some of the ideas that we would have like to study are:

- Can we mitigate the effects of the catastrophic forgetting problem through vocabulary extensions?
- Does this method work for context-aware embeddings?
- Can we have an Online NMT model that learns its vocabulary on the fly?

Acknowledgements. Work supported by the Horizon 2020 - European Commission (H2020) under the SELENE project (grant agreement no 871467), and the project Deep learning for adaptive and multimodal interaction in pattern recognition (DeepPattern) (grant agreement PROMETEO/2019/121). We gratefully acknowledge the support of NVIDIA Corporation with the donation of a GPU used for part of this research.

References

1. ACL: Ninth workshop: Statistical machine translation (2014)
2. ACL: First conference: Machine translation (wmt16) (2016)
3. ACL: Fifth conference on machine translation (wmt20) (2020)
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2015)
5. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. CoRR abs/1607.04606 (2016)
6. Carrión, S., Casacuberta, F.: Autonmt: a framework to streamline the research of seq2seq models (2022). https://github.com/salvacarrion/autonmt/
7. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on EMNLP, pp. 1724–1734 (2014)
8. Conneau, A., Schwenk, H., Barrault, L., LeCun, Y.: Very deep convolutional networks for natural language processing. CoRR abs/1606.01781 (2016)
9. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018)
10. Gillick, D., Brunk, C., Vinyals, O., Subramanya, A.: Multilingual language processing from bytes. CoRR abs/1512.00103 (2015). http://arxiv.org/abs/1512.00103
11. Ha, T., Niehues, J., Waibel, A.H.: Effective strategies in zero-shot neural machine translation. CoRR abs/1711.07893 (2017). http://arxiv.org/abs/1711.07893
12. Johnson, M., et al.: Google's multilingual neural machine translation system: enabling zero-shot translation. CoRR abs/1611.04558 (2016). http://arxiv.org/abs/1611.04558

13. Koehn, P., et al.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL 2007, pp. 177–180 (2007)
14. Kudo, T.: Subword regularization: improving neural network translation models with multiple subword candidates. In: Proceedings of the 56th Annual Meeting of the ACL (Volume 1: Long Papers), pp. 66–75 (2018)
15. Liu, T., Ungar, L., Sedoc, J.: Continual learning for sentence representations using conceptors. ArXiv abs/1904.09187 (2019)
16. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on EMNLP, pp. 1412–1421 (2015)
17. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2–4, 2013, Workshop Track Proceedings (2013)
19. Neubig, G., Watanabe, T., Mori, S., Kawahara, T.: Substring-based machine translation. Mach. Transl. **27**(2), 139–166 (2013)
20. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on ACL. ACL 2002, pp. 311–318 (2002)
21. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation, pp. 1532–1543
22. Peters, M.E., et al.: Deep contextualized word representations. CoRR abs/1802.05365 (2018)
23. Post, M.: A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 186–191 (2018)
24. Qi, Y., Sachan, D., Felix, M., Padmanabhan, S., Neubig, G.: When and why are pre-trained word embeddings useful for neural machine translation? In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), June 2018
25. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers), pp. 1715–1725 (2016)
26. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) NIPS, vol. 27 (2014)
27. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the 31st NeurIPS. NIPS 2017, pp. 6000–6010 (2017)
28. Vilar, D., Peter, J.T., Ney, H.: Can we translate letters? In: Proceedings of the Second WMT. StatMT 2007, pp. 33–39 (2007)
29. Xu, H., Liu, B., Shu, L., Yu, P.S.: Lifelong domain word embedding via meta-learning. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, pp. 4510–4516, July 2018
30. Xue, L., et al.: Byt5: towards a token-free future with pre-trained byte-to-byte models. CoRR abs/2105.13626 (2021). https://arxiv.org/abs/2105.13626

31. Zhang, B., Williams, P., Titov, I., Sennrich, R.: Improving massively multilingual neural machine translation and zero-shot translation. CoRR abs/2004.11867 (2020). https://arxiv.org/abs/2004.11867
32. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: evaluating text generation with BERT. CoRR abs/1904.09675 (2019)