Cross-Lingual Sentence Embedding: Building Equivalent Representations in Italian and French through a Unified Latent Space

Deborah Dore
University of Rome *La Sapienza*
dore.1994616@studenti.uniroma1.it

Abstract

This paper presents a revised technique to generate cross-lingual sentence embeddings through contrastive approaches. A downstream task, translation, will be used to evaluate how well the embeddings perform. The model will be able to extract the necessary information from the embedding of a sentence in a language it wasn't trained on if the embeddings of the languages are sufficiently close, and the translation process will go smoothly.

1 Introduction

Embeddings are fixed-length, dense representations. They can be used to represent data such as *words*, *sentences* and *documents*. In the context of Natural Language Processing, *word embeddings* are used to represent words in a format that we can easily manipulate. Multiple techniques have been proposed for converting natural language text into numerical representations, including one-hot encoding, term frequency-inverse document frequency (TF-IDF), and more advanced methods like Word2Vec, GloVe and transformer-based models like BERT.

Neural Machine Translation (NMT) has exploded in recent years, leading to the creation of numerous tools deployed all over industry today [13]. Recent approaches often make use of the *attention mechanism* to emphasize the critical elements within a sentence, with the objective of making the translation more effective. Encoder-Decoder attention-based architectures like BERT have accomplished major improvements in machine translation. However, many of these approaches focus on the translation of English to/from another language, largely due to the numerous resources for the English language.

Cross-lingual embeddings aim is to remove language barriers by representing several languages with the same data, hence avoiding the need for different language-specific models for different languages. This type of embedding can be generated using several techniques. We need to differentiate between the **comparability** and the **type of alignment** in order to categorize them[12].

- *Type of alignment*: specifies the type of bilingual supervision signal. *Word, sentence, or document level* can be used.
- *Comparability*: information can be *parallel* if it is an exact translation or *comparable* if it is only somewhat similar.

The goal of this research is to develop an autoencoder that generates cross-lingual embeddings at the sentence level by utilizing parallel data. Cross-lingual embeddings can be evaluated in a multitude of tasks. The downstream tasks chosen in this system to evaluate the effectiveness of the embeddings is translation.
With this goal in mind, two autoencoders - one for each language - are trained. Every encoder develops the ability to produce embedding that resemble their counterparts in other languages. Each decoder learns how to reconstruct the original sentence from the embedding. Note that each model sees just sentences in one language throughout the training tasks. The translation task will be carried out by feeding the Italian decoder with the embedding generated by the French encoder, or vice versa, making use of the similarities between the French and Italian embeddings. As a result, we will get the French sentence's translated into Italian.

Unlike other systems, the one detailed in this paper seeks a more *reusable* NMT approach by utilizing cross-lingual embeddings. In this setup, obtaining a Spanish-Italian translation simply requires training an encoder to create Spanish embeddings resembling those of the Italian encoder, and then utilizing the pre-trained Italian decoder for translation.

1.1 Examples

This system employs two autoencoder models, one per language. We will distinguish them as follows: *encoder_it* refers to the encoder responsible for generating low-dimensional embeddings of Italian sentences, while *encoder_fr* performs the same task for the French sentences. Similarly, *decoder_it* refers to the decoder responsible for reconstructing Italian sentences and *decoder_fr* reconstructs French sentences. The goal of this study is being able to swap the decoders after training and obtain good translation, thus proving the goodness of the generated cross-lingual embeddings. Applying *decoder_it* to the embedding generated from *encoder_fr* will result in a translation from French to Italian.

- Translation Italian to French:

  decoder_fr(encoder_it(*Buongiorno. Tutto bene?*)) = *Bonjour. Tu vas bien?*

- Translation French to Italian:

  decoder_it(encoder_fr(*Comment est le temps aujourd'hui?*)) = *Oggi che tempo fa?*

1.2 Real-world applications

Translation systems are used worldwide for tasks such as translating web pages, emails, or social media posts, enabling communication across different languages. The most popular translation system used today is Google Translate provided by Google. Other as powerful recent applications are DeepL or Bing Translator. All of them support a wide range of languages and are accessible from the web or from the specific mobile application.

2 Related work

Exploiting the power of embeddings that are similar in different languages, *cross-lingual embeddings*, has always been an hot topic. In their work, *Kim et al.*[8] present a denoising autoencoder that has been trained on dataset of monolingual text. They obtain state-of-the-art performance in English-German translations by using adversarial strategies to generate cross-lingual embeddings. This type of embeddings can also be useful for other downstream tasks. *Chandar et al.*[6] demonstrate how a model trained on a source language (e.g. English), using cross-lingual embeddings, is able to classify documents in another target language (e.g. German). Similar to what this paper wants to achieve, *Yang et al.*[16] have presented a system that maps pairs of sentences from different languages to a shared-latent space, enhance the cross-language translation using GANs leading to significant improvements on English-German, English-French and Chinese-to-English translation tasks. Furthermore, *Deng et al.*[14] make use of a variational autoencoder in their approach using only sentence level parallel corpus.

3 Datasets and benchmarks

Due to the popular nature of this tasks, dataset and benchmarks are relatively easy to locate. Some of the most common dataset are:

- **Workshop on Machine Translation (WMT) Datasets**[1] that includes translation datasets for various language pairs
- **International Workshop on Spoken Language Translation (IWSLT) Datasets**[2] that contains datasets for speech translation tasks
- **European Parliament Proceedings Parallel Corpus**[9], a parallel corpus extracted from the proceedings of the European Parliament

The most popular datasets used to benchmark machine translation systems belongs to the WMT family. Up to date benchmark show that the best results are achieved using Transformer models, evaluated using BLEU score.

dataset	model	bleu score
WMT2014 English-French	Transformer+BT	46.4
IWSLT2014 German-English	BiBERT	38.61
WMT2014 English-German	Transformer Cycle	35.14
WMT2016 English-Romanian	DeLighT	34.7

Table 1: Benchmarks from the PapersWithCode Leaderboard

4 Existing tools, libraries, papers with code

Given that Transformers are the predominant architecture for this task, the *HuggingFace Transformer Library*[15] stands out as the most comprehensive resource, ensuring access to an extensive amount of pre-trained transformer models. To construct and train models from scratch, the *PyTorch*[11] and *Tensorflow*[3] libraries are the most used. These libraries offer support and flexibility for those who wish to build things from the bottom up. To clean and process the dataset, the Natural Language Toolkit (NLTK)[5] is the common choice. Of the work previously cited, we can also find an implementation by *Yang et al.*[16] in Python using Tensorflow[1].

In this work, NLTK and Pytorch are the primary tools used to respectively preprocess the dataset and create the models from scratch.

5 State-of-the-art evaluation

Neural Machine Translation models are mostly evaluated using automatic evaluation metrics such as **BLEU** and **METEOR**. The *Bilingual Evaluation Understudy*[10] measures the overlap of n-grams between a candidate corpus (obtained as output from the model) and the reference translations (usually more than one, the *real* translations). It calculates the similarity between the reference corpus and the candidate corpus. Among precision-based implementations in the MT evaluation, BLEU is the most widely used metric.[7] The *Metric for Evaluation of Translation with Explicit ORdering*[4] considers precision, recall, stemming, synonymy, and word order to evaluate translations. METEOR calculates a score based on the alignment of words between the reference corpus and candidate corpus.

6 Experimental Settings

Concerning the implementation of the system described in this paper, the dataset used is the *European Parliament Proceedings Parallel Corpus*. In particular, its *english-to-french* and its *english-to-italian* corpus. The sentences contained in these two corpus were aligned to form an *french-italian* parallel corpus. It's worth noting that the two corpora did not have the same number of sentences due to different translations. In fact, during the European Parliament session, not all languages are translated to english and then translated back to the target language. Some languages are directly translated from source to target without going through english first. This creates misalignment in the corpus. Therefore, some loss of information is expected.

[1]GitHub repository

Two models, one per language, were trained jointly. Each model is composed of an encoder and a decoder. Encoder and Decoder share a latent space.

- The encoder is composed of an Embedding Layer and 3 LSTM layers. It takes as input a sentence and outputs a low dimensional representation of that sentence and its hidden states.

- The shared latent space is composed of one linear layer followed by the Rectified Linear Unit (ReLU) activation function. This layer is shared between encoder and decoder. It takes as input the low dimensional representation produced by the encoder and outputs a low dimensional embedding.

- The decoder is composed of 3 LSTM layers and a Linear Layer. It takes as input the embedding and the hidden states produced respectively by the latent space and the encoder and outputs a fixed-length representation of the sentence. For each word in the output, the model generates a probability distribution over the original vocabulary. Therefore, instead of directly predicting the next word, the model provides a probability for each word in the vocabulary, indicating the likelihood of it being the next word in the sequence. The loss is calculated over the probabilities. During sentence generation, the word that corresponds to the index in the vocabulary with the highest probability is selected.

If we want to generate cross-lingual embeddings between Italian and French, we have to train two autoencoders simultaneously. The loss will be a sum of three losses: a negative log likelihood loss between input and output of each decoder that ensures that the decoder learns how to reconstruct the sentence from the embedding (Equation 1)

$$\mathcal{L}_{\text{NLL}}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}), \qquad (1)$$

where:

N is the number of samples,

C is the number of classes,

y_{ij} is the indicator function for the true class of sample i being j,

$\hat{y}_{ij}$ is the predicted probability of sample $i$belonging to class j.

and a contrastive loss calculated over the embeddings generated by the shared latent space that ensures that embeddings of sentences with the same meaning are positioned close in the space (Equation 2).

$$\begin{aligned}\mathcal{L}_{\text{contrastive}}(x_1, x_2, y) = y \cdot \mathrm{D}(x_1, x_2)^2 \\ +(1-y) \cdot \max(margin - \mathrm{D}(x_1, x_2), 0)^2\end{aligned} \tag{2}$$

where:

$D(x_1, x_2)$ is the distance metric between x1 and x2,

y indicates whether the samples are a +/- pair,

$margin$ is the minimum desired distance between negative pairs.

To fine-tune the positioning of sentence representations in the latent space, the contrastive loss approach makes use of both positive and negative pairs. Its main objective is to drive "negative" pairs apart while simultaneously bringing related or "positive" pairs close together. Positive pairings in this sense are made up of two sentences in different languages with the same meaning. Conversely, a negative pair consists of two sentences in different languages with different meaning. The goal of this approach is to enforce a clear distinction between sentences that have dissimilar interpretations and to guarantee that sentence with the same meaning are clustered closely in the learnt space.
Figure 1 explain the behaviour of the model during training.

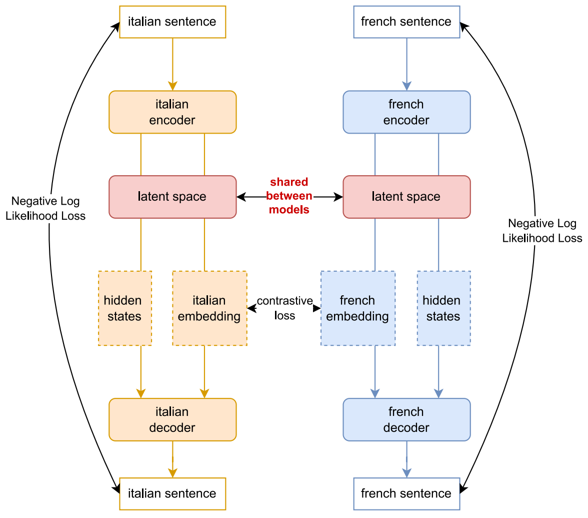

Figure 1: System's behaviour during training

The contrastive loss employed is crucial for the model's good outcome. If the encoders learns to produce effective cross-lingual embeddings, we could easily exchange the encoders and decoders to perform translation. Figure 2 explains how to perform translation from French to Italian after model's training.

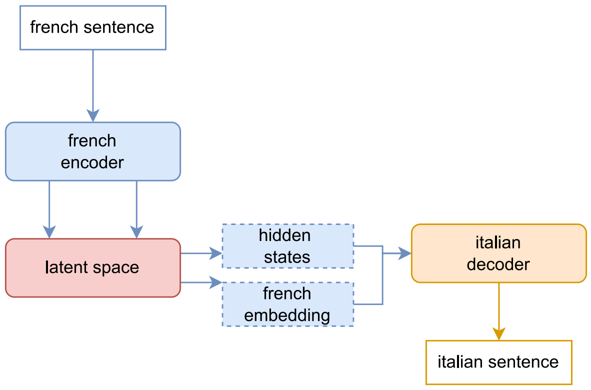

Figure 2: System's behaviour during translation

To encourage the model to pay more importance to the goal of creating similar embeddings, the contrastive loss was specifically scaled with regard to the sum of the two NLL[2] Losses like showed in Equation 3

$$\begin{aligned}\mathcal{L}_{\text{total loss}}(y, \hat{y}) = \alpha \cdot \mathcal{L}_{\text{contrastive}}(x_1, x_2, y) + \\ \beta \cdot (\mathcal{L}_{\text{NLL italian}}(y, \hat{y}) + \mathcal{L}_{\text{NLL french}}(y, \hat{y}))\end{aligned} \tag{3}$$

[2]Negative Log Likelihood

6.1 Results

The model's alignment of the embeddings was sufficiently satisfactory after 30 epochs of training. Nevertheless, the decoders appear to be incapable of translating sentences, even if the encoders have learned to align words closely together. Despite their confidence in reassembling the sentences they have been instructed on, they have not received enough instruction to be able to do so given the opposing language's embedding. Results are display in Table 2.

Dataset	Translation Direction	BLEU	METEOR
WMT2014	fr $\rightarrow$ it	0.0	0.0033
WMT2014	it $\rightarrow$ fr	0.0	0.0007

Table 2: Best results of the experiments

Despite all efforts to adjust the architecture and components such as optimizers and learning rate schedulers, the model was unable to perform translation. As a result, this remains an open issue. The system was also trained on french-english and german-english translation to compare it to state-of-the-art models and benchmarks, however it shown no ability to carry out translation.

dataset	model	bleu score
WMT2014 English-French	Transformer+BT	46.4
WMT2014 English-French	Noisy back-translation	45.6
WMT2014 English-French	mRASP+Fine-Tune	44.3
WMT2014 English-French	Transformer + R-Drop	43.95
WMT2014 English-French	**Autoencoder**	0.0

Table 3: Benchmarks from the PapersWithCode Leaderboard

dataset	model	bleu score
WMT2014 English-German	Transformer Cycle	35.14
WMT2014 English-German	Noisy back-translation	35.0
WMT2014 English-German	Transformer+Rep	33.89
WMT2014 English-German	T5-11B	32.1
WMT2014 English-German	**Autoencoder**	0.0

Table 4: Benchmarks from the PapersWithCode Leaderboard

6.1.1 Hyper-parameter optimization

Research on the model's most crucial hyper-parameters, including learning rate, batch size, number of layers in both the encoder and decoder's LSTM, size of the embeddings and layers, and dropout in the LSTM layers, has been carried out in order to improve the model's effectiveness. Furthermore, distinct weights were assigned to the different parts of the loss.

The results of the optimization are listed in Table 5.

hyper-parameter	best	search space
batch size	**16**	discrete: [16, 32, 64, 128]
learning rate	**0.0002**	loguniform: [1e-4, 1e-1]
encoder embedding dim	**150**	discrete: [50, 100, 150]
encoder lstm hidden dim	**256**	discrete: [64, 128, 256]
latent space dim	**16**	discrete: [16, 32, 64]
LSTM num layers	**2**	discrete: [1, 2, 3]
encoder's dropout	**0.139**	loguniform: [0.1, 0.3]
decoder's dropout	**0.102**	loguniform: [0.1, 0.3]
alpha	**1.039**	loguniform: [0.5, 1]
beta	**1.440**	loguniform: [0.5, 1]

Table 5: Result of the hyper-parameter optimization

The optimization result indicates that in order for the model to learn shared representations, there needs to be a tiny bottleneck and a significant amount of information compression (from a size 256 representation to a size 16 representation). Specifically, a higher priority was given to the reconstruction loss than was expected during the loss's design. A simple justification would be that aligning embeddings in a shared space is not nearly as complex as reconstructing a high dimensional text from a smaller embedding representation.
Figure 3 illustrates the model's loss decrease during training using the optimal hyperparameters.

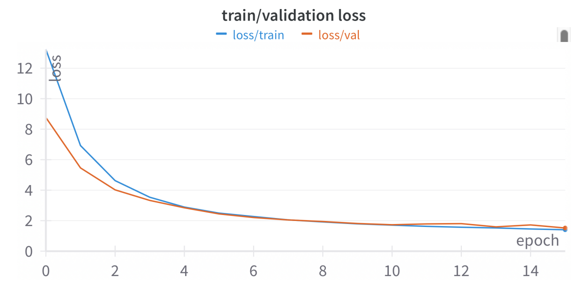

Figure 3: Training and Validation loss decrease during training

In particular, in Figure 4 we can observe the decrease of the different parts of the loss during training.

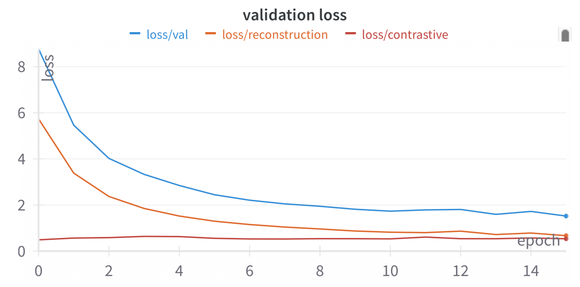

Figure 4: Contrastive and Reconstruction loss decrease during training

When compared to the reconstruction loss, we can see that the contrastive loss decreases significantly more slowly. As a result, one might think about employing two separate learning rates for the parameters controlling the recontruction loss and contrastive loss.

6.1.2 Ablation Study

An ablation study was carried out, using the newly created French-Italian parallel dataset, in order to have a better understanding of the model's key features and challenges.To determine what specifically contributes most to the model's training, various aspects of the model were eliminated in each test, including dropout from the LSTM layers, the use of contrastive loss, the use of reconstruction loss, and several layers within the LSTM.

Avoiding to apply dropout in every LSTM layer had little effect on the model's performance, but employing just one layer undoubtedly limited the model's capacity to reconstruct the sentences.
By training without contrastive loss, the embeddings were prevented from aligning in the latent space, which nullified the model's objectives. Remarkably, it also had an impact on reconstruction loss, which began to rise continuously for five epochs and eventually caused the model to stop early due to the early stopping mechanisms in place. This outcome could be explained by contrastive loss confining the embeddings into a smaller space than in the absence of it. As a result, the model faced more difficulty learning a sparser distribution than it would have with a denser embedding distribution. In Figure 5 we can see the projection of the latent space created by the model with all features. The latent space of the Italian sentences and the French sentences match.

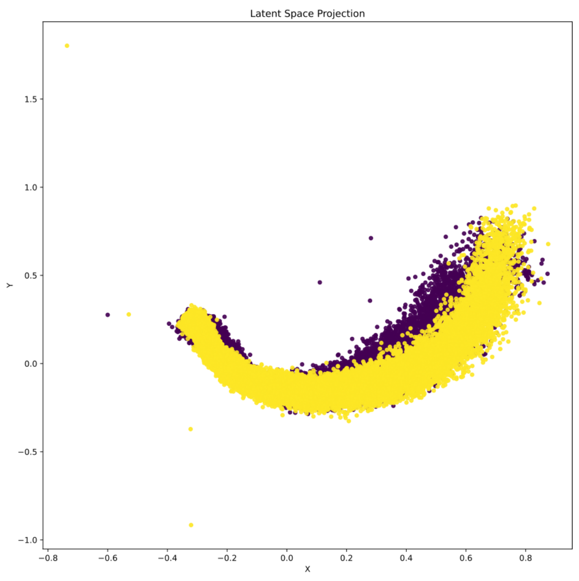

Figure 5: Latent Space Projection

When reconstruction loss was not included in the model's training procedure, the decoders became ineffective. They were unable to recreate phrases because they had not been trained to reproduce the original sentence. The embeddings did move closer together when the contrastive loss was applied, but the model's effectiveness was compromised and it was unable to serve its intended function.

The outcomes of the study are listed in Table 6. For simplicity, the scores listed below are the result of translation from French to Italian only.

MODEL TYPE	METEOR	BLEU	FINAL LOSS
Only Reconstruction Loss	0.009	0.0	442828.058
Only Contrastive Loss	0.002	0.0	21.145
Zero Dropout	0.003	0.0	7.536
One LSTM layer	0.006	0.0	7.480
All Features	0.003	0.0	1.520

Table 6: Results of the ablation study

6.2 Discussion

There can be multiple reasons for why this system is not working properly. One problem can be attributed to the architecture: errors in the construction of the architecture can prevent the model from learning. Other problems could be attributed to the contrastive loss not being *strict* enough. Other autors[8][16] have preferred the use of Adversarial Training to force a certain behaviour in the model: the contrastive loss alone could not be sufficient to out final purpose. The translation task could also be to ambitious as a downstream task for such an architecture, classification task such as in [6] could be more easily achievable than translation tasks.

Making the bottleneck of the model more restrictive helped it keep only the most important information. Adjusting the reconstruction and contrastive loss magnitudes slightly improved how the model understood tasks. The decoder successfully reconstructed sentences during training using embeddings and hidden cells, showing good results.
One idea worth exploring further is using the encoder's hidden state for the decoder. It seems the decoders might rely too much on hidden states and overlook the importance of embeddings. When switching both embeddings and hidden states for translations, the decoders seem to struggle, indicating a need to balance the emphasis on hidden states versus embeddings for better performance.

7 Conclusions

The objective of this study is to present a novel architecture that uses contrastive approaches to produce cross-lingual embeddings between Italian and French. In order to accomplish this, the European Parliament Proceedings Parallel Corpus was used to create a new parallel sentence-aligned French-Italian corpus. For each language, an individual model is created. Sentences from the two languages are projected onto a shared latent space, where sentences are arranged as closely as possible employing a contrastive loss. The model then attempts to rebuild the sentences starting from the low dimensional representation of the latent space using a Negative Log Likelihood Loss. The goodness of the embedding is evaluated on a downstream task, translation, using METEOR and BLEU score as metrics. To perform translation, the decoders are simply swapped.

Although the model performs well when aligning sentences closely together and reconstructing sentences to their original dimensions, it encounters challenges in translating sentences when the decoders are switched.
This limitation, however, presents opportunities for further development. To investigate this idea, we could try to employ two distinct optimizers. To prevent erroneous training of distinct portions, we should have one optimizer for the reconstructive loss and another for the contrastive loss.
Additionally, a sequential training strategy—initially training the encoders jointly and then the decoders independently—may stabilize learning by enabling the decoders to learn insights from the embeddings in a more structured manner.
Another improvement is the use of independent linear layers for each encoder, which allows them to learn language-specific word distributions before aligning the space using contrastive loss applied to the output of each layer.

The code is publicly accessible in the corresponding GitHub repository. The results of the trainings are accessible in the corresponding Weights&Biases Report.

References

[1] 2020 Fifth conference on Machine Translation (WMT20). 2

[2] International Conference on Spoken Language Translation (IWSLT). 2

[3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow, Large-scale machine learning on heterogeneous systems, November 2015. 3

[4] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 3

[5] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python:*

Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Inc., 2009. 3

[6] Sarath Chandar AP, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. *Advances in neural information processing systems*, 27, 2014. 2, 7

[7] Eirini Chatzikoumi. How to evaluate machine translation: A review of automated and human metrics. *Natural Language Engineering*, 26(2):137–161, 2020. 3

[8] Yunsu Kim, Jiahui Geng, and Hermann Ney. Improving unsupervised word-by-word translation with language model and denoising autoencoder. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 862–868, 2018. 2, 6

[9] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit 2005*, pages 79–86, 2005. 2

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002. 3

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 3

[12] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631, 2019. 1

[13] Felix Stahlberg. Neural machine translation: A review. *Journal of Artificial Intelligence Research*, 69:343–418, 2020. 1

[14] Liangchen Wei and Zhi-Hong Deng. A variational autoencoding approach for inducing cross-lingual word embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4165–4171, 2017. 2

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 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, October 2020. Association for Computational Linguistics. 2

[16] Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. Unsupervised neural machine translation with weight sharing. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–55, Melbourne, Australia, July 2018. Association for Computational Linguistics. 2, 3, 6