# A Survey on Unsupervised Neural Machine Translation

**Yunzhe Wang, Yunzhe Zhang, Ganghua Mei**
Department of Computer Science
Columbia University
New York, NY 10027
{yw3737,yz4197,gm3044}@columbia.edu

## Abstract

Unsupervised Neural Machine Translation (UNMT) is a sub-field of machine translation that involves training translation models without using parallel data. In recent years, UNMT has gained significant attention due to the prevalence of monolingual data and the availability of large-scale language models. The building blocks of UNMT consist of two procedures: 1) Word embedding pre-training, and 2) Latent Space Alignment. The former involves creating dense work representations on a large corpus of monolingual data. The latter involves aligning the embedding spaces of different languages and creating a synthetic dictionary that enables word-to-word translation completely unsupervised. Later techniques such as Back-Translation enable alignment at the sentence level, achieving state-of-the-art performance.

## 1 Introduction

Neural Machine Translation (NMT) refers to the use of artificial neural networks to translate text from one language to another. It has achieved significant performance in recent years due to its ability to handle complex and variable sentence structures, and to improve the quality of translations compared to traditional statistical machine translation methods[2, 25, 8, 24]. NMT typically involves large-scale neural networks trained on large parallel sentence-pair datasets in different languages. Nonetheless, parallel corpora are scarce resources for most language pairs, while it's not the case for monolingual data, which are much more available for each language. Recent studies in Unsupervised Neural Machine Translation (UNMT) point toward a promising direction for machine translation research, and even improving translation model performance in the absence of parallel data [7, 15, 14].

Most successful techniques for unsupervised machine translation require two steps. First, to separately pre-train source and target languages for word embeddings to learn a transformation function that allows the alignment of word embedding. The aligned word embedding creates a synthetic dictionary between the two languages, where words with the same meaning can be matched through nearest neighbor search, thus enabling unsupervised word-to-word translation. Techniques such as Back-Translation[22] then allow alignment at the sentence level and improve translation performance beyond word-to-word.

**Survey Outline** The survey paper consists of two main sections introducing the main procedures of UNMT. The first part illustrates various techniques and neural network architectures for producing word embeddings, including matrix factorization, skip-gram models, and de-noising auto-encoders. The second part mainly explains various techniques that achieve the embedding alignment of different languages, including dictionary supervision methods, adversarial training, and back-translation.

Preprint. Under review.

Different methods will then be compared based on some benchmark datasets. Lastly, challenges in this field will be outlined, and future directions will be proposed.

## 2 Word Embedding Pre-training

Word embedding represents words in a continuous vector space, where semantically similar words are mapped to nearby points in the space. Most successful techniques for learning word representations are based on the linguistic distributional hypothesis of Harris[13], which states that a word that appears in the same context conveys a similar meaning.

One surprising aspect of deriving word embeddings through neural networks is that the resulting word vectors exhibit approximately additive compositionality[17, 19, 12]. Adding two word vectors together often results in a vector closest to the vector representing the composite of the added words. For example, adding the vectors for "man" and "royal" results in a vector that is closest to the vector for "king." This property allows us to use these vectors to answer word analogy questions using algebraic methods. For instance, to answer the question "Man is to king as woman is to ?" one can return the word whose vector is closest to the vector obtained by subtracting the "man" vector from the "king" vector and adding the "woman" vector.

In this section, we will introduce three different types of techniques that produce word embeddings based on the distributional hypothesis.

### 2.1 Matrix Factorization Methods

**Latent Semantic Analysis** Latent Semantic Analysis (LSA) [9], aka Latent Semantic Indexing (LSI) in the context of information retrieval, is a technique used to analyze relationships between a set of documents and the terms they contain by producing a matrix of "concept" scores. By identifying these shared contexts, LSA can uncover the underlying concepts that the words represent, even if those concepts are not directly mentioned in the text. The technique uses singular value decomposition (SVD) to reduce the dimensionality of the term-document matrix, representing the relationships between the terms and documents. The dimensionality reduction result is a set of concept scores representing the text's latent, or hidden, concepts.

**Latent Dirichlet allocation** Latent Dirichlet allocation (LDA) [3] is a statistical model that is used to uncover the underlying topics in a corpus of text. It does this by representing documents as mixtures of topics, where each topic is distributed over the words in the vocabulary. LDA assumes that the words in each document are generated from a mixture of a small number of topics and that each word is generated from a mixture of the topic distributions for that document. To apply LDA to a corpus of text, we first need to construct a term-document matrix, which represents the relationships between the terms and documents in the corpus. This matrix is then decomposed using matrix factorization techniques, such as singular value decomposition (SVD) or non-negative matrix factorization (NMF). This decomposition results in a set of topic distributions, representing the underlying topics in the corpus.

**Global Vectors** Global Vectors for Word Representation (GloVe)[20] is a model for distributed word representation that uses unsupervised learning to map words into a space where the distance between them is related to their semantic similarity. Training is done using global word-word co-occurrence statistics from a corpus, and the resulting word vectors exhibit linear substructures. GloVe is a log-bilinear regression model that combines the benefits of global matrix factorization and local context window methods.

### 2.2 Local Context Window Methods

Local context window methods are techniques used in natural language processing to model the meaning of a word in a given sentence. These methods use the surrounding words in a sentence to determine the context in which a given word is used, and then use that context to generate a more accurate representation of the word's meaning.

For example, consider the sentence "The cat sat on the mat." In this sentence, the word "cat" is used in a specific context, namely that it is a small, furry animal. A local context window method would

use the words "The," "sat," "on," and "mat" to determine that the word "cat" is being used to refer to a cat, and would generate a more accurate representation of the word's meaning based on this context.

Local context window methods are often combined with other natural language processing techniques, such as word vector representation, to create more accurate models of word meaning. These methods can be applied to various natural language processing tasks, including sentiment analysis, named entity recognition, and machine translation.

**Word2Vec** Word2Vec [18] uses a shallow neural network with a single hidden layer. The input to the network is a one-hot encoded vector representing the input word, and the output is a set of probabilities for each word in the vocabulary. The hidden layer contains the word vectors, which are learned during training.

Two main algorithms are used in word2vec: the continuous bag-of-words (CBOW) model and the skip-gram model. The CBOW model tries to predict the current word given the surrounding words, while the skip-gram model tries to predict the surrounding words given the current word. Both algorithms can be effective, but the skip-gram model is generally considered to perform better with larger datasets.

**FastText** FastText[4] is a similar word representation learning techniques as compared to Word2Vec. It is particularly useful for working with large text datasets because it is computationally more efficient. Unlike Word2Vec, FastText uses hierarchical softmax, a faster method for training word vectors than the traditional softmax approach. Also, FastText uses sub-word information, which allows it to create vectors for out-of-vocabulary words by combining the vectors of their component subwords. This makes FastText effective at handling rare words and words with spelling variations. For example, the embedding of the word "cat" would be represented as a single vector if using Word2Vec, whereas FastText uses a combination of the character n-grams "c," "ca," "cat," "a," "at," and "t."

### 2.3 Denoising Auto-Encoding

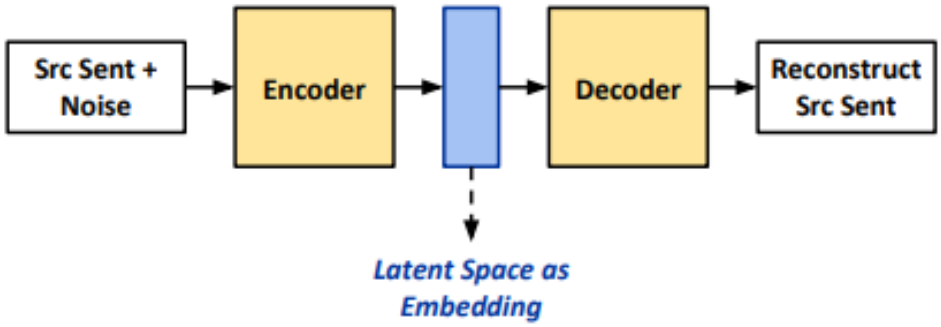


Figure 1: Note: Src Sent + Noise in the Figure represents a corrupted version of a sentence from the target language after some operations including Token Masking, Token Deletion, and Token Permutation

A de-noising autoencoder (DAE) is trained to reconstruct a clean "repaired" input from a corrupted version of it. This is done by first corrupting the initial input $x$ into $\tilde{x}$ by means of a stochastic mapping. Common text noises include:

- **Token Masking:** Randomly replace word tokens in a sentence with a mask token `[MASK]`. The task is commonly known as Masked Language Modeling (MLM).
- **Token Deletion:** Randomly delete word tokens in a sentence. Note the difference between token masking and token deletion is that the former preserves sentence length while the latter does not.
- **Token Permutation:** Randomly shuffle word orders in a sentence.

Corrupted input $\tilde{x}$ is then mapped to a latent representation $y = f_\theta(\tilde{x})$ from which we reconstruct a $z = g_{\theta'}(y)$. Parameters $\theta$ and $\theta'$ are trained to minimize the average reconstruction error over a training set to make $z$ as close as possible to the uncorrupted input $x$. State-of-the-art de-noising methods such as XML[14], BERT[10], BART[16], mostly utilizes the Transformer architecture[24] with Masked Language Modeling.

## 3 Latent Space Alignment

Mikolov observed that the structures of continuous word embedding spaces are similar across languages, including those that are distantly related, such as English and Vietnamese. They proposed using linear mapping from a source to a target embedding space based on this similarity.

To learn this mapping, they used a parallel vocabulary of 5,000 words as anchor points and evaluated their approach to a word translation task. Since then, various efforts have been made to improve cross-lingual word embeddings, such as the iterative method of Artetxe [1], which reduces the needed dictionary size to as little as 25 words. Nonetheless, all of these alignment methods rely on bilingual word lexicons.

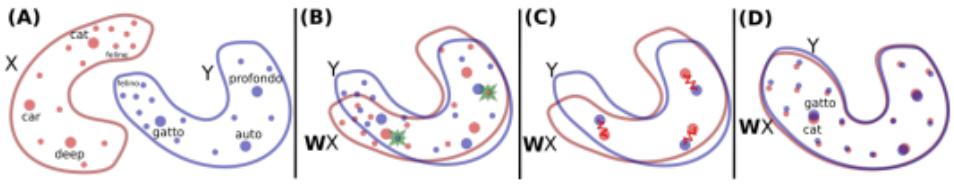


Figure 2: (A) Suppose we have two distributions of work embeddings: English and Italian. We denote English words as X and Italian words as Y. Our goal is to learn a W to align or translate these words. Within this embedding, each dot represents a word in its corresponding language. The size of the words is proportional to their frequency during the training step. If the size of the dot is large, this word frequently appears when we train the corpus of that language. (B) Using the adversarial learning introduced in section 3.2, we learn a matrix W so that WX could be as close to Y as possible to align these word embeddings. The green stars marked in the graph are discriminators that aim to determine whether the two different word embeddings come from the same distribution. If the discriminators trained from adversarial learning cannot further distinguish the two word embeddings, then we know the two embeddings have been aligned together as much as possible. Our end goal in adversarial thus is to confuse the discriminator. (C) Unfortuantely, the W learned in (B) cannot outperform the supervised methods in most cases. To further refine W, we used the Procrustes algorithm by using frequent words aligned in the previous step as anchor points to minimize the energy function between these anchor points.[21] (D) Finally, we aim to translate the English words to the Italian words within the embedding (e.g., from X to Y) using the learned mapping W.

Recent attempts at creating cross-lingual word embeddings without any parallel data utilize Adversarial approaches. The aligned word embeddings create a synthetic dictionary that enables word-to-word translation completely unsupervised. Figure 4 illustrated this idea with a toy example.

More advanced techniques that utilize sequence models, such as LSTM, and iterative Back-Translation, enable the alignment of latent embedding at the sentence level.

### 3.1 Adversarial Domain Adaptation

Adversarial domain adaptation is a machine learning technique that involves adapting a model trained on one dataset (the source domain) to perform well on another dataset (the target domain) that has a different underlying distribution [23]. This is often necessary because, in many real-world applications, the data available for training a model may not represent the data the model will encounter when deployed in the real world.

The key idea behind adversarial domain adaptation is to use an adversarial training process, in which two neural networks are trained simultaneously: a feature extractor and a domain discriminator (see

figure 3). The feature extractor network is trained to extract useful features from the input data, while the domain discriminator is trained to predict the domain (source or target) of the input data.

During training, the feature extractor is given input data from both the source and target domains, and it is trained to extract features that are useful for predicting the correct output in both domains. Meanwhile, the domain discriminator is trained to predict the domain of the input data, using the features extracted by the feature extractor network as input.

Formally, the discriminator has trained to classify the language source by minimizing the cross-entropy loss: $|_D(\theta_D|\theta, Z) = -E_{(x_i,l_i)}[\log p_D(l_i|e(x_i, l_i))]$, where $(x_i, l_i)$ is the sentence and language id pairs sampled from the two monolingual datasets, $\theta_D$ is the discriminator parameters, $\theta_{enc}$ is the encoder parameters, and $Z$ are the encoder word latent embeddings.

The encoder and the decoders are trained to reconstruct the original sentence:

$$L_{auto}(\theta_{enc}, \theta_{dec}, Z, l) = E_{x\sim D_l, \hat{x}\sim d(e(C(x),l),l)}[\Delta(\hat{x}, x)]$$

The encoder is also trained to fool the discriminator

$$L_{adv}(\theta_{enc}, Z|\theta_D) = -E_{(x_i,l_i)}[\log p_D(l_j|e(x_i, l_i))]$$

with $l_i = l_1$ if $l_i = l_2$, and vice versa.

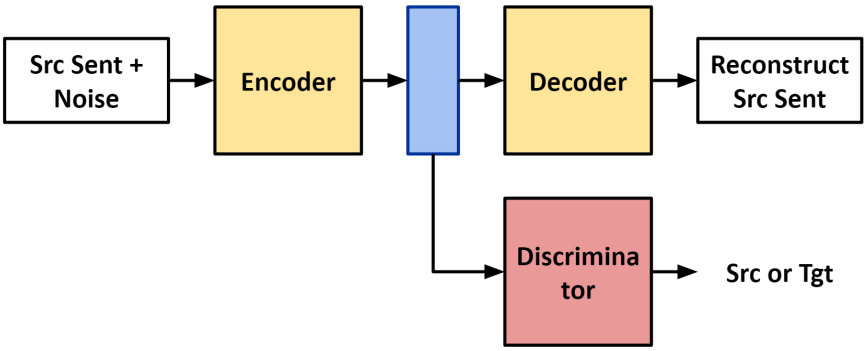


Figure 3: Diagram of Adversarial Approach. The discriminator's objective is to classify the source of the latent embedding. The encoder's training objective is to reconstruct the corrupted sentence and to confuse the discriminator

### 3.2 Back-Translation

Back-Translation infers to provide the monolingual training data with a synthetic source sentence that is obtained by automatically translating the target sentence into the source language. It pairs the monolingual training sentences with the corresponding synthetic source sentence from which the context vector could be derived. Its further implementation will be shown in the next section Iterative algorithm.

### 3.3 Iterative Algorithm

Guillaume (2018) [15] proposed the Iterative algorithm, mainly consisting of two architectures: Denoising Auto-Encoding and Back-Translation, mentioned in section 2.3 and section 3.3, respectively. As shown in Figure 3, the upper part stands for the process of the Denoising Auto-Encoding, and the lower part represents the Back Translation

The logic behind the Iterative algorithm is described as follows which concentrates on the sequence-sequence translation. This algorithm highly relies on an iteration process starting from an initial translation model $M^1$ in the pseudocode line 3. This $M^1$ is learned through the achievement

from the model proposed by Conneau (2017)[7] and thus derived M could be used to translate the available monolingual data, which is a necessary step in back-translation. Next, the for loop in lines 4-8 will update the encoder and decoder with the parameters needed and minimize their objective function. In the end, $M^{t+1}$ consisting of the newest encoder and decoder will be returned. The specific process will be shown below.

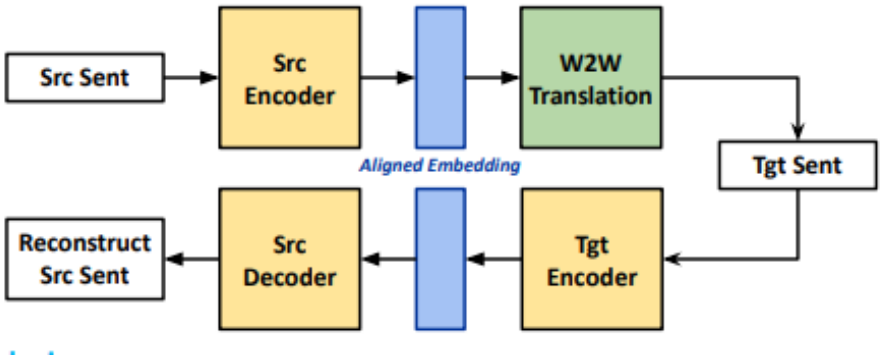


Figure 4: Explanation of the proposed Iterative algorithm. This model is a sequence-to-sequence architecture which means it could achieve sentence translation after training instead of word to word any more. This architecture contains a decoder and encoder trained from the two different separate languages. Suppose the model has already finished the auto-encoding part by learning to denoise the sentence in each language domain. This figure is a translation part similar to auto-encoding except for encoding from another language. This architecture utilizes the input of the translation model produced in the previous interaction (upper part of this graph) and back translates the Tgt Sent on the right side to go through the Tgt encoder and Scr decoder again. In the end, by comparing the Src Sent and Reconstruct Src Sent, the objective function will be further optimized by trying to align the two latent spaces shown in the figure. (W2W translation is a synthetic dictionary obtained from the model proposed by Conneau (2017) [7]

**Algorithm 1** Unsupervised ways to do the Machine Translation

1: **function** TRAINING($D_{src}$, $D_{tgt}$, T)
2: Build a bilingual synthetic dictionary using the monolingual data from section 3.2
3: $M^1 \leftarrow$ building word-to-word machine translation using the dictionary in the previous step
4: **for** t$\leftarrow$ 1; t $<$ T; t++ **do**
5: using updated $M^t$ to translate the monolingual dataset
6: discriminator training and model training
7: $\theta \leftarrow$ arg min $L_D, \theta_{enc}, \theta_{dec}, Z \leftarrow$ arg min L
8: $M^{t+1} \leftarrow e^t d^T$ ▷ update the MT model
9: **end for**
10: **return** $M^{t+1}$

(Note: $D_{src}$ stands for the monolingual dataset of the source language and $D_{tgt}$ stands for the monolingual dataset of the target language. T represents the total iterations that the decoder and encoder will be updated. $\theta_{discr}, \theta_{enc}, \theta_{dec}, \theta dec$ represents the parameter of the discriminator, encoder, and decoder. Z represents the embedding of the monolingual dataset. Lastly, L means the objective function defined by Guillaume (2018) [15].

# 4 Methods Evaluation

## 4.1 Benchmark Dataset

We identified several benchmark datasets commonly used for machine translation models training and evaluation. In particular, we considered two language pairs: English-French and English-German. Note that English-French is a closer language pair than English-German.

**WMT 2014** The WMT 2014 dataset [5] is a collection of datasets used in shared tasks of the Ninth Workshop on Statistical Machine Translation, featuring tasks including a standard news translation task, a separate medical translation task, a task for run-time estimation of machine translation quality, and a metrics task.

**WMT 2016** The WMT 2016 dataset [6] is a collection of datasets used in shared tasks of the First Conference on Machine Translation, featuring tasks including five machine translation (MT) tasks (standard news, IT-domain, biomedical, multimodal, pronoun), three evaluation tasks (metrics, tuning, run-time estimation of MT quality), and an automatic post-editing task and bilingual document alignment task.

**Multi30k-Task1** task 1 of the multi30k dataset [11] has 30,000 images, with English, French, and German annotation that are translations of each other. The parallel annotation of different languages can be treated as a dataset for translation evaluation.

## 4.2 Unsupervised Model Selection Criterion

In unsupervised settings, it is difficult to choose the best hyper-parameters, especially for the discriminator, decoder, and encoder, using some criterion since there is no access to the parallel sentences to determine the quality of the proposed model. Guillaume proposed a criterion that highly correlates with the BLEU Score, a highly used metric in the NLP field to overcome this difficulty. This newly proposed metric is based on the translation quality from the forward direction and the backward direction mentioned. It calculates the BlEU score via this two-step translation process and is averaged over these two directions. Then the parameter with the highest score will be adopted.

Suppose there are two monolingual datasets $D_{src}$ and $D_{tgt}$ for different languages, an encoder e, and a decoder d. Then denote $M_{src\to tgt}(x) = d(e(x, src), tgt)$ as the translation model in the forward direction and $M_{tgt\to src}(x)$ the backward-direction. The metric will be denoted as $MS(e, d, D_{src}, D_{tgt})$ which could be formulated as:

$$\begin{aligned} MS(e, d, D_{src}, D_{tgt}) = &\frac{1}{2} E_{x\sim D_{src}} BLEU[(x, M_{src\to tgt} \circ M_{tgt\to src}(x))] \\ &+ \frac{1}{2} E_{x\sim D_{tgt}} BLEU[(x, M_{tgt\to src} \circ M_{src\to tgt}(x))] \end{aligned}$$

This metric could achieve two goals. (1) This metric could determine the T to stop training. (2) This metric could also select the best hyper-parameters based on the model performance since it measures the quality of translation in both directions with equal importance using the BLEU score.

## 4.3 Evaluation

In this section, we compared the three UNMT methods 1) word-level alignment by Conneau[7], 2) Sentence-level alignment through LSTM denoising autoencoder by Lample[15], and 3) Sentence-level alignment through the Transformer architecture and masked language modeling by Lample[14]. Evaluation data are taken from the three papers on the WMT English-French and English-German WMT datasets, with the BLEU score metric. See table 1 for the result.

## 4.4 Limitations and Challenges

Many existing works focus on modeling UNMT systems, but there is a lack of research on why UNMT works and the scenarios where it is effective. In particular, UNMT still has limited performance when dealing with distant language pairs, domain-specific scenarios, and efficiency.

| | en-fr | fr-en | de-en | en-de |
|---|---|---|---|---|
| word-by-word | 6.28 | 10.09 | 10.77 | 7.06 |
| sentence lstm | 25.1 | 24.2 | 17.2 | 21.0 |
| sentence transformer (XLM) | 33.4 | 33.3 | 26.4 | 34.3 |

Table 1: BLUE score evaluation of three UNMT methods: 1) word-level translation, 2) Sentence-level translation through LSTM, and 3) Sentence-level translation through Transformer. Evaluation data are taken from the three papers on the WMT English-French and English-German WMT datasets.

**Distant Languages** The performances of UNMT commonly do not perform better in distant language pairs such as Chinese/Japanese-English than similar language pairs such as German/French-English. This could be due to two reasons:

1. The syntactic structures of distant language pairs are very different. Therefore, without parallel supervision, syntactic correspondence is very difficult to learn for UNMT.
2. There may not be enough shared words/subwords in the distant language pair to learn the shared latent representation.

**Efficiency** Compared to NMT, UNMT has a rapid increase in training time. Furthermore, sharing latent representations between the two translation directions can affect performance, especially for distant language pairs. Denoising can also slow convergence by continuously modifying the training data. The efficient training of UNMT is a major issue that needs to be addressed.

## 5 Conclusion

This paper thoroughly discusses two main aspects of Unsupervised Neural Machine Transition(UNMT): Word Embedding Pretraining and Aligning Embedding Latent Space. Word Embedding Pre-training essentially constructs a vector space where points around the same area denote semantically similar words. The main insights of this procedure are built upon Harris' distributional hypothesis. This paper discusses three methodologies that produce word embeddings built upon this hypothesis: Matrix Factorization Methods (e.g., Latent Semantic Analysis, Latent Dirichlet allocation, Global Vectors), Local Context Window Methods (e.g., Word2Vec, FastTexT), and Denoising Auto-Encoding. Latent Space Alignment essentially constructs a linear mapping between source and target embedding space based on the insights that the structures of continuous word embedding spaces are similar across languages. Specifically, UNMT leverages Adversarial approaches, such as Adversarial domain adaptation, Back-Translation, and Iterative Algorithm, to create cross-lingual word embeddings using monolingual data, so that the aligned word embeddings create a synthetic dictionary that enables word-to-word and subsequently sentence to sentence translation.

This paper then discusses the significant performances of UNMT on several benchmark datasets, including WMT 2014/2016 and Multi30k-Task1, and touched on the evaluation criteria, such as Guillaume's criterion and the BLUE score. This paper concludes with the potential limitations of UNMT, such as distant languages and Efficiency, and extrapolates the future direction of this line of research.

## References


[1] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. "Learning bilingual word embeddings with (almost) no bilingual data". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017, pp. 451–462.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[4] Piotr Bojanowski et al. "Enriching word vectors with subword information". In: *Transactions of the association for computational linguistics* 5 (2017), pp. 135–146.

[5] Ondřej Bojar et al. "Findings of the 2014 workshop on statistical machine translation". In: *Proceedings of the ninth workshop on statistical machine translation*. 2014, pp. 12–58.

[6] Ondřej Bojar et al. "Findings of the 2016 conference on machine translation". In: *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. 2016, pp. 131–198.

[7] Alexis Conneau et al. "Word translation without parallel data". In: *arXiv preprint arXiv:1710.04087* (2017).

[8] Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. "A survey of multilingual neural machine translation". In: *ACM Computing Surveys (CSUR)* 53.5 (2020), pp. 1–38.

[9] Scott Deerwester et al. "Indexing by latent semantic analysis". In: *Journal of the American society for information science* 41.6 (1990), pp. 391–407.

[10] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[11] Desmond Elliott et al. "Multi30K: Multilingual English-German Image Descriptions". In: *Proceedings of the 5th Workshop on Vision and Language*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 70–74. DOI: 10.18653/v1/W16-3210. URL: http://www.aclweb.org/anthology/W16-3210.

[12] Alex Gittens, Dimitris Achlioptas, and Michael W Mahoney. "Skip-gram- zipf+ uniform= vector additivity". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017, pp. 69–76.

[13] Lawrence Harris. "Transaction data tests of the mixture of distributions hypothesis". In: *Journal of financial and Quantitative Analysis* 22.2 (1987), pp. 127–141.

[14] Guillaume Lample and Alexis Conneau. "Cross-lingual language model pretraining". In: *arXiv preprint arXiv:1901.07291* (2019).

[15] Guillaume Lample et al. "Unsupervised Machine Translation Using Monolingual Corpora Only". In: *International Conference on Learning Representations*. 2018.

[16] Mike Lewis et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension". In: *arXiv preprint arXiv:1910.13461* (2019).

[17] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* 26 (2013).

[18] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[19] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. "Linguistic regularities in continuous space word representations". In: *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*. 2013, pp. 746–751.

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

[21] Peter H Schönemann. "A generalized solution of the orthogonal procrustes problem". In: *Psychometrika* 31.1 (1966), pp. 1–10.

[22] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Improving neural machine translation models with monolingual data". In: *arXiv preprint arXiv:1511.06709* (2015).

[23] Eric Tzeng et al. "Adversarial discriminative domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7167–7176.

[24] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[25] Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. "A survey of deep learning techniques for neural machine translation". In: *arXiv preprint arXiv:2002.07526* (2020).