EasyChair Preprint

№ 11990

Word-Alignment Emphasized Dependency Aware Decoder (WDAD) with Data Augmentation in Nonautoregressive Translation

Yupei Li

February 8, 2024

Imperial College
London

MSc Individual Project

Imperial College London

Department of Computing

Word-alignment emphasized Dependency Aware Decoder (WDAD) with Data Augmentation in Nonautoregressive Translation

Author:
Yupei Li

Supervisor:
Professor Lucia Specia &
Mr Nihir Vedd

Second Marker:
Dr. Chiraag Lala

Submitted in partial fulfillment of the requirements for the MSc degree in Computing (Artificial Intelligence and Machine Learning) of Imperial College London

Feb 2024

Abstract

The Non-Auto-Regressive model (NAT) for machine translation offers increased efficiency compared to autoregressive models but faces challenges related to target-side dependencies. Two issues arise: over and under-translation; and a multi-modal problem of natural language. To mitigate these problems, previous researchers have made extensive efforts, particularly with the Dependency Awareness Decoder (DAD) model. While these models focus on retaining target-side dependencies to enhance performance to some extent, they still leave two gaps in cross-lingual translation tasks: word embeddings in shared embedding space and shared character sequences. This paper proposes two solutions to address these issues, namely adaptation from the Ernie-M model and data augmentation involving language BPE(LBPE), respectively. Additionally, the paper explores their combined effect, enabling language prompts to help the model distinguish tokens from different languages and cluster words from a semantic perspective. Thus, the Word-alignment Language-Prompted DAD (WDAD) model with data augmentation is proposed, which indeed demonstrates progress.

Combination model of LBPE and CAMLM contributes approximately +0.5 BLEU score points on the WMT14 De-En pair dataset, and CAMLM contributes approximately +1 BLEU score points on the WMT16 En-Ro dataset, while the combined model exhibits limitations in its interaction with the combined work due to the inappropriate data augmentation strategy of LBPE, as evidenced by a mixed data strategy and language embedding layer, and the baseline data augmentation strategy. But this does not deny the principle of LBPE and any effects LBPE made at all. It is just a sign that there are better solutions for data augmentation strategy. Additionally, the combined model faces challenges in the word clustering issue arising from contradictions in traditional encoding strategies in translation and CAMLM. To address this, the paper proposes an idea with conducting unfinished experiments, leaving it for the future.

Acknowledgements

This work is for the MSc program. I would like to express my sincere gratitude to Dr. Nihir Vedd who gave me tremendous technical supervision and instructions as well as motivation for my research. Also, I want to thank Professor Chiraag Lala and Lucia for giving an overall evaluation of my work. Moreover, I want to thank my family for giving me mental support during this work.

Contents

Chapter 1

Introduction

1.1 Motivations

The Machine Translation (MT) task, as discussed in [1], is a critical issue in the field of Natural Language Processing (NLP). It involves the automatic translation of text from one natural language to another, and it has seen rapid advancements alongside the development of deep learning models. In the beginning, rule-based models[2] were the first translation models based on linguistic rules. They retrieved token information, including semantics and grammar, from a dictionary and used it for translations. However, designing effective rules is time-consuming, and a significant amount of linguistic information needs to be inserted manually. Phrase-based models[3], also known as statistical methods, apply the Bayes Theorem to decode sentences. They are more efficient than the rule-based models but may have errors in the translation results. As deep learning develops, numerous neural networks have been applied to address the challenges of Neural Machine Translation (NMT), with certain models even capable of facilitating translation between multiple languages. NMT includes the Cross-lingual Machine Translation (XMT) task, which has witnessed rapid growth [4], along with related tasks like cross-lingual information retrieval, driven by the diverse array of languages spoken around the world. Typical methods include RNN and LSTM[5], which learn information for entire sentences but may introduce errors in handling long-term dependencies. The attention mechanism resolves this issue by calculating an attention matrix over the entire sentence. Therefore, Transformer models[6] based on the attention mechanism utilize the attention mechanism to address this issue.

Transformer models [6] are currently the most widely used methods for NMT and XMT. The base transformer includes an encoder that maps the source language to an embedding space and a decoder that decodes auto-regressively from another embedding space to the target language, word by word. The attention mechanism is used to learn the connections and semantic similarities between the sentence itself and sentences from language pairs. It allows for long-dependencies and serves as the foundation of large language models such as Bert[7], which achieve outstand-

ing performance on XMT and related word embedding tasks. Furthermore, it is universally applicable for tasks in NLP such as Question & Answer, Named Entity Recognition, and so on.

The classical transformer is an auto-regressive translation(AT) transformer model, which excels in NMT performance compared with numerous deep learning methods, such as RNN, LSTM[5]. Target tokens are decoded sequentially, implying a one-by-one generation approach. Subsequent tokens draw upon information from previously decoded tokens, leading to time-consuming processing, particularly for long target sentences.

In contrast, non auto-regressive translation(NAT) methods are designed to expedite decoding by generating target tokens simultaneously in one pass. Multiple methods demonstrate acceleration speeds of up to 21 times[8].

However, NAT methods sacrifice the decoder's accuracy because they fail to capture the dependencies from target languages [9]. In other words, NAT needs to tackle the problem of lack of information and instructions given by the target side. This causes two main drawbacks in NAT performance. One is over-translated (repeatedly translated) and under-translated (neglected translated). For example, *thank you* might be translated into *Danke Danke*, where *thank* was over-translated and *you* was under-translated. The NAT would output consecutive repeated tokens or not output some of the important tokens because the model is not aware whether a certain word has been decoded, therefore it would keep decoding one word and occupy the whole sentence length limitation. Consequently, some words would be translated many times leaving others no space. But this could be avoided by AT since the previously decoded tokens would instruct the later decoding together with the output of the embedding. The other is the *multi-modal*[9] translation problem, where the natural language is flexible and uses different expressions to convey the same meaning. **Note that the term *multi-modal* here does not refer to the conventional meaning of multi-modality, such as the combination of images and texts. Instead, in the context of the NAT field, it represents a specific expression denoting the presence of multiple choices or potential states during decoding, reflecting linguistic diversity.** Therefore, there would be many correct translation formats for one sentence. For example, *Vielen dank* could be either translated into *many thanks* or *thank you* but the NAT model would have difficulties in the alignment of the words´ translations. Therefore, different supervisions, which are the multiple different correct translation targets used to calculate the loss, would confuse the NAT model, to make both of the words aligned to *thank*, while the AT model would be able to see the context and deterministically generate the target sentence.

NAT boasts impressive efficiency, but there is still room for improvement in terms of accuracy with a notable performance decrease (approximately 6 BLEU[10] score decrease in NMT task) compared to AT, as demonstrated in a study by Ren et al. [11]). Therefore, this research aims to design strategies to address the challenges arising from target-side dependencies.

To alleviate the problem of a lack of internal dependencies in the target language,

significant efforts are made in many aspects. The different types of strategies are Iterative Refinement[12], Data Distillation[13], Learning Strategies, Loss Changes and so on (more details in Section 2.1). The Iterative Refinement process takes the output of the decoder as the input for the next iteration, which can be time-consuming and may diminish the advantages of NAT. In contrast, one-pass NAT models decode the target sentence once and achieve the theoretically maximum decoding speed. Notably, the Dependency-Aware Decoder (DAD) focuses on one-time generation of targets[14]. It uses filtering to capture similarities between source and target languages and uses a carefully designed three-phase training process to better capture the context. (More details in Section 2.2). This is the core base model that motivates the research presented in this report.

Despite contributing significantly to improving NAT models, DAD still has two gaps: first, words tend to cluster by their meaning within the same language, but it would be ideal if they clustered independently of language; and second, the same character sequences carry different meanings in different languages. More specifically, words from different languages have distinct vector representations within the same embedding space. Distinguishing them saves memory and helps cross-lingual token learning. However, word representations from the same language often tend to cluster together applying conventional XML training strategy. [15], which contradicts our expectation that words conveying the same meaning should cluster together. Additionally, many cross-lingual corpora share numerous identical character sequences that convey disparate meanings but might be mistakenly recognized in their meaning in NMT. For instance, with regard to English and French, *ant* in English is a variant of *anti*, signifying *before* (e.g., *antibody*), while in French, it pertains to specific agents (e.g., *enseignant*).

These two issues are considered in XLM training models[16], and this work will focus on addressing these issues. Two approaches are motivated for the two issues respectively. First, the XLM pretraining model utilizes language embedding to understand cross-lingual corpora better. At the same time, Ernie-M[17] uses a special attention strategy to prevent **information leakage**, which refers to the exposure of linguistic features within the same language, which has the potential to recluster words based on language rather than solely on semantic factors. This process forces the model to learn not only cross-lingual semantics but also language-specific structures Drawing inspiration from Ernie-M[17], a component akin to Cross-Attention Masked Language Modeling (CAMLM) is integrated into the original DAD model. This enables the model to rely solely on tokens from the other language when learning word embeddings from a single language token. In other words, this strategy facilitates easier word alignment in terms of semantic meaning. This approach compels the model to cluster words from diverse languages, as it lacks language information for its original language. CAMLM has proven its utility [17], across various NLP tasks, such as Cross-Lingual Natural Language Inference (XNLI), Named Entity Recognition (NER), and in NMT tasks as well. Second, a novel format for Byte-pair Encoding (BPE)[18] tokens is designed to distinguish identical character sequences originating from different languages. This is inspired by data augmentation

strategies, where manipulating data significantly influences the performance of the model.By appending a distinct language indicator to BPE tokens, the downstream process can readily discern the language of the token, thereby enlarging the size of the vocabulary and providing better capabilities for word representation. Therefore, the model would not be confused about the meaning of one token if it represents different meanings for different languages. This not only enhances the effectiveness of the attention mechanism in DAD but also aids in distinguishing the diverse semantic meanings conveyed by the same character sequence.

1.2 Contributions

To overcome the challenges for NAT and XML as presented above, this report presents five main contributions:

Improve word representation DAD employs a shared embedding space and a joint shared dictionary for language pairs, aiming to acquire clearer relations between the source and target languages. However, it is sub-par because the words in shared embedding space tend to cluster by their meaning within one language due to the training strategy. To mitigate the problem, rather than directly utilizing Bert-base parameters for sentence tokenization and word embedding extraction, this study integrates components resembling CAMLM into DAD. This integration serves to group words that convey identical semantic meanings within the embedding space. Section 3.2 of the report presents these components, while Section 3.3 explains how the two models are combined. The results show that the CAMLM component could decrease the distance of the word embedding cluster centre in the shared embedding space, thus enhancing the final result.

Word alignment pre-trained strategy The CAMLM component demonstrates its ability to provide new word embeddings. However, traditional DAD training strategies could impair this capability, as discussed further in Section 4.5. To address this issue, this study introduces a pre-training strategy that explicitly supervises the model training using a provided word alignment dictionary. This approach ensures that the model places emphasis on avoiding the reclustering of word embeddings based on language during training. Details of this method are presented in Section 3.2.

Mix data set strategy Training CAMLM with DAD might compromise the effects of the CAMLM-like module. This could occur because CAMLM training involves encoding processes for a multilingual corpus, whereas DAD training, a translation task, focuses solely on encoding the source-side language. DAD may potentially segregate words based on the source and target languages without explicit indication. Hence, this work proposes a strategy to mix data so that multiple languages can coexist on both the target and source sides. For instance, in an En-De task, English could be considered both the source and target language within the same training epoch. This approach aims to alleviate the interference caused by the dominance of a single language on the same side for CAMLM.

Design data augmentation strategy: Language-specific BPE(LBPE) tokens The study aims to enhance the original BPE tokens by incorporating a distinct language signal from its source language. This addition seeks to mitigate ambiguity arising from identical character sequences across different languages. The introduced Language-Based BPE (LBPE) contributes to improved token differentiation within the shared embedding space, particularly when dealing with more than two languages. At the same time, it doubles the vocabulary size compared to the original one, thereby increasing the number of parameters to be trained in the embedding layer and endorsing the model more capability to extract language features. Further discussion on this topic will be provided in Section 3.2. Our results show that LBPE has **+0.5** BLEU score improvement.

Design Language Embedding layer Though LBPE plays a role in distinguishing tokens from different languages, using LBPE would significantly increase the parameter size by a large margin (twice as large in the WMT14 dataset). Consequently, this increase could introduce extra changeable variables and more training time. This work introduces a language embedding layer inspired by LBPE but reduces the parameter size back to its original dimensions while maintaining the effects of the LBPE to prove whether LBPE would be an appropriate data augmentation strategy.

1.3 Report Structure

The report is divided into 5 chapters. Chapter 2 describes a comprehensive overview of the theoretical background most relevant to this work. This includes a comparison between NAT and AT, along with their related technical background involving Transformers (Section 2.1), a detailed technical structure of DAD (Section 2.3), a detailed explanation of the masked language model CAMLM (Section 2.4), and an overview of the shared embedding space (Section 2.5).

The critical methodology of the thesis is introduced in Chapter 3. It outlines the integration of the two approaches with DAD and also the mixed data strategy with the language embedding layer. Chapter 4 presents the evaluation and relevant comparisons among the various integration methods. Lastly, Chapter 5 highlights the results and outlines potential future research directions. This research project has no ethical issues.

1.4 Logistics

This research is motivated by a fundamental problem in NAT, specifically the target-side dependency. DAD is a classical method that applies NAT in the XML field, addressing some superficial issues such as over-under translation and multi-modal problems. However, DAD encounters issues with word embedding and shared character sequences. Therefore, this research initially proposes CAMLM and data augmentation, specifically LBPE to address these issues individually and conducts experiments on each strategy separately and in combination.

During experimentation, these two strategies perform well following with theoretical analysis, since the main issues are DAD issues, but limitations are observed for both. Subsequently to make the models better, three additional ideas are proposed: a language embedding layer, word alignment supervision, and a combination of all (WDAD). The initial experiments reveal that LBPE is not the best choice for data augmentation but plays a role in distinguishing shared character sequences. Ongoing research is being conducted for the remaining two ideas.

Chapter 2

Background and related works

This section will delve into the technical background and related literature. The primary focus of this work lies in the NAT model applied to translation tasks, particularly emphasizing enhancements in word embeddings within a shared embedding space. To begin, the work will review the widely recognized translation model, Transformers[6], serving as the foundational model adaptable to AT and NAT strategies. With knowledge of the technicalities in the base model, a careful comparison between NAT and AT will be conducted to elucidate the challenges inherent in NAT for translation, particularly highlighting the word alignment issue due to the lack of target side dependency in the decoding process of Transformers. Following this analysis of word alignment issues, attention will shift towards exploring solutions within the realm of shared embedding spaces to address the word embedding and shared character sequences challenges faced in NAT-based translation models.

2.1 Transformers

The traditional transformer[6] is shown in Fig2.1. It involves an encoder and decoder with main components with multi-head attention and cross attention mechanism grasping the connection between language pairs. Subsequently, after getting $X' = Embed(X) + Pos$ adding positional encoding to word embedding, whose shape is $\mathbb{R}^d$, where d is the hidden dimension of the encoder, it would perform multi-head attention as follows:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}\left(\text{head}_1, \ldots, \text{head}_\text{h}\right) W^O \\ \text{where } head_i &= \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \end{aligned} \quad [6]$$

where $W_i^Q \in \mathbb{R}^{d_{model} * d_q}, W_i^K \in \mathbb{R}^{d_{model} * d_k}, W_i^V \in \mathbb{R}^{d_{model} * d_v}$ and $W_i^O \in \mathbb{R}^{hd_v * d_{model}}$ are parameters and d_q, d_k, d_v are hidden dimensions of query, key and value respectively, and h is the number of head. $Q \in \mathbb{R}^{S * d_{model}}, K \in \mathbb{R}^{d_{model} * S}, V \in \mathbb{R}^{S * d_{model}}$, where S is the sequence length, are gained from X' together with parameters W_q, W_k, W_v. Then the output of multi-head attention is passed to the residual network and layer normalization, and then to the feed forward network. After N stacks of the same

structure to get the embedding of the source tokens, the final output of the encoder X_{emb} is fed into the cross attention mechanism of the decoder. The decoder's attention mechanism follows the same principle as the multi-head attention in the encoder, but it uses masks on subsequent tokens during decoding to maintain sequential order. Particularly, there is cross attention mechanism in the decoder that has the same structure but the Q, K, V is produced by Y_{hid}, W_q, X_{emb}, W_k, X_{emb}, W_v respectively, where Y_{hid} is the current hidden state of the decoder. This aims to incorporate the encoder outputs. The decoder has N stacks as well. Then it would experience the *softmax* layer to output the target tokens word by word.

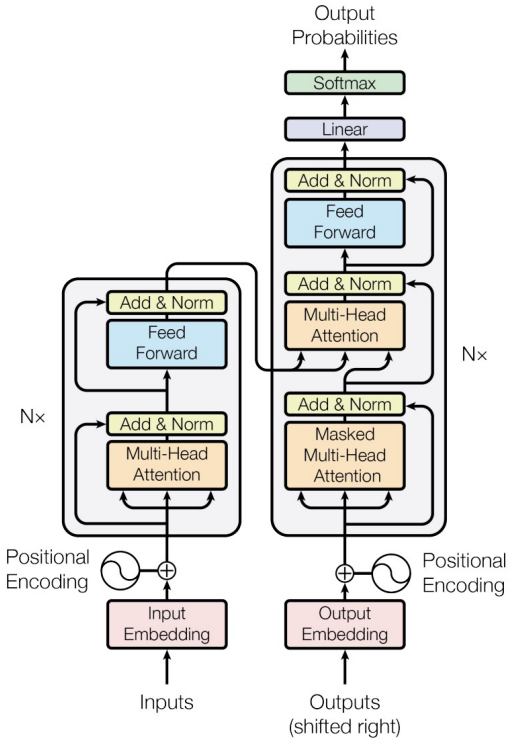

Figure 2.1: Transformer architecture[6]

The transformer has been widely used in previous work and proved to be effective in NMT tasks for its ability to capture the relation between long sequences.

2.2 NAT and AT

With knowledge of the transformer, the base, and the state-of-the-art model used in NAT and AT, this section will compare the differences between NAT and AT using the transformer in the translation task to explain the reason for problems raised in NAT.

2.2.1 AT technical background

The traditional AT model can be formulated as a sequential generation problem given the source language sequence $X = \{x_1, x_2, ..., x_N\}$ as a condition. The target tokens y_i in target language sequence $Y = \{y_1, y_2, ..., y_M\}$ refer to X and the previous target generated tokens $y_1, y_2, ..., y_{i-1}$. The objective of the task is to maximize the likelihood:

$$L = \prod_{i=1}^{T} P(y_i|X, y_{<i}; \theta) \tag{2.1}$$

where θ represents the AT model (commonly a neural network). It clearly shows that when decoding y_i, the model relies on both the source language and the previously decoded tokens that would gain much target dependency but it cannot be parallelised since it generates in a sequential order.

2.2.2 NAT background

When compared to AT, NAT enjoys broad application prospects due to its higher efficiency, as seen in various areas such as Automatic Speech Recognition (ASR) [19], Dialogue Generation [20], Semantic Parsing [21], and numerous generative tasks in NLP. Furthermore, it plays a significant role in Neural Machine Translation (NMT) tasks [9], as it can generate all target tokens in parallel. These strategies can be applied to existing deep learning models such as transformers [22]. This would significantly reduce decoding time compared to traditional AT, although it presents a challenge in capturing target-side dependencies, also known as internal dependencies from the target side.

The challenge in principle is that NAT would not rely on any previously decoded tokens but instead generate the target words all in one go. The objective of the task is accordingly to maximize the likelihood:

$$L = \prod_{i=1}^{T} P(y_i|X; \theta) \tag{2.2}$$

where θ represents the NAT model (commonly a neural network) and T is the length of the target sentences but need to be predicted in the model. T could be formulated as below:

$$T(\theta) = P(L|X, \theta) \tag{2.3}$$

where more straightforwardly, it aims to learn parameters to map from the length of the source sentence to that of target length L [23].

Therefore, the inference strategy is slightly different compared with AT. NAT has no mask and output all the target tokens while AT does have access to the latter tokens when inferring and only rely on previous tokens. Consequently, the previously decoded tokens are put in decoding again to be considered in AT but there is only an initial state in NAT.

Therefore, if the NAT model is used, critical problems include numerous translation challenges such as the multi-modal problem defined earlier and over/under-translation [9]. More specifically, since the NAT does not consider previously decoded tokens as a condition, generating sentences coherently and ensuring word alignment becomes more challenging for the decoder. The generation process may become fragmented and repetitive. Additionally, having multiple sources of supervision could confuse the NAT model because it might select sub-sequences of the sentence, and align them with related target token outputs as translation results. However, the model combines them without knowing which token has already been decoded so there could be repeated alignment. Due to the different references and supervisions, several words may be output repeatedly while others remain untranslated.

These issues are widely researched in the literature. The next subsection will discuss previous approaches to deal with these issues.

2.2.3 Previous approaches

Since many challenges stem from the loss of target-side dependency, previous efforts have made significant attempts to restore target-side dependency. This subsection summarizes the work aimed at mitigating the dependency issue.

The initial solution was introduced in Jiatao Gu etc.'s research [24] and was built on traditional Transformer models. This solution incorporated a *fertility* module, consisting of a one-layer network with a softmax layer, serving as a latent variable to offer additional contextual information from the entire sentence, shown in Fig2.2. This module provides external cues for aligning word pairs in the language. ´Fertility' refers to the number of times that the meaning of certain words could appear in the decoding output. This restriction helps narrow the output distribution and significantly reduces the generation of repetitive consecutive sequences during the decoding process.

Mathematically, the loss function of the whole model is shown in 2.4

$$p_{\mathcal{NA}}(Y \mid X; \theta) = \sum_{f_1,\ldots,f_{T'} \in \mathcal{F}} \left(\prod_{t'=1}^{T'} p_F\left(f_{t'} \mid x_{1:T'}; \theta\right) \cdot \prod_{t=1}^{T} p\left(y_t \mid x_1\left\{f_1\right\}, .., x_{T'}\left\{f_{T'}\right\}; \theta\right) \right) \tag{2.4}$$

where p_F is the fertility probability and p is the common NAT probability.

As a result, this approach leads to a 1.5-point improvement in BLEU score over WMT16 data set translation while achieving a 15-fold increase in speed. However, this method is associated with a single challenge. Despite its success, there remain multiple valid translation possibilities and variations in reference outputs, which is the multi-modal problem.

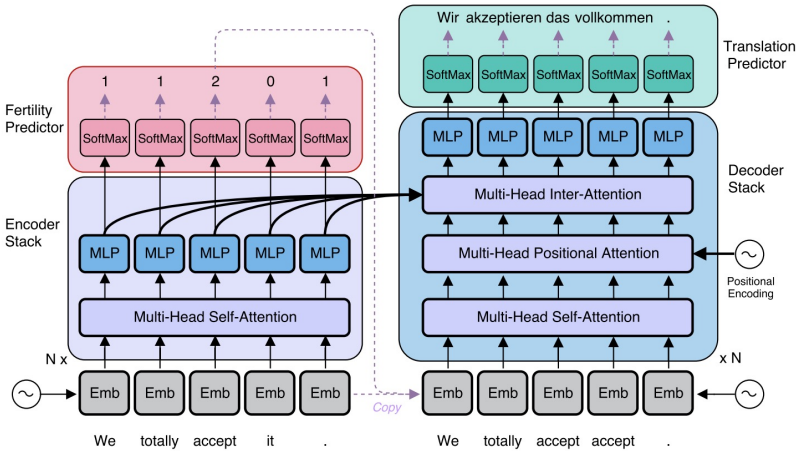

Figure 2.2: Initial solution[24]

There are other aspects of strategies to make up for the two issues followed up by the original model. Some of them are trying to solve the issue such as multi-modality directly - Knowledge Distillation, others may focus on the principle problem that leads to the two issues - target side dependency.

Knowledge Distillation (KD) [13] proves to be a helpful approach in tackling the multi-modal problem. KD involves preprocessing the dataset ($[S, T'']$) using an AT model, generating the target language (T') with the source language (S) as input. The output T' is deterministic and machine-generated, making it more easily recognizable by the machine. Consequently, the new language pair $[S, T']$ exhibits less noise and fewer multi-modal problems. An example is that the translation of *Vielen Danke* is *thank you*, instead of several other translation options like *many thanks*.

Data Distillation is one **data augmentation** strategy. Other data augmentation methods include synonym replacement [25], sentence shuffling [26], and more. Feng's team [27] summarized approaches to strengthen the data by incorporating syntactic information [28], using an external dictionary for more parallel data [29], and others. However, these data augmentation strategies require an external model to process the relevant linguistic features. There are also non-traditional data augmentation methods with the same purpose: to expand the training dataset and enhance the model's representation capability. Simultaneously, increasing the number of parameters can result in a larger language model with more linguistic features, as evidenced in experiments[30]. Consequently, expanding the vocabulary size, which is another non-traditional method of data augmentation that could potentially improve performance due to the increased diversity of tokens and enhanced representation ability, is selected in this research.

Some other approaches are designed to better capture the target side dependencies. **Latent Variable**: Given that the initial solution for NAT with latent variables contributes to an increase in BLEU score, there are concepts for the Latent Variable method that focus on extracting additional conditions from the source side. Nader

Akoury's team [31] group the source tokens according to their syntactic information. These groups then serve as additional conditions that restrict the generated tokens in terms of syntax, thus narrowing the potential for over-translation. Chitwan Saharia's team [32] uses a latent variable as prior alignment probability and applies it for CTC and Importer loss. A sequence of latent variables aids in aligning words with each other, reducing the likelihood of over-translation. Also, target side dependency could be represented either semantically or syntactically. **DAD model** would be the core method and also the latent-variable method. Yu Bao's team[33] designed a quantified vector as a target categorization code, which has a similar role to Part-of-Speech (POS) tags, to explicitly emphasize the target side syntactic information. Though the tags are fuzzy, they instruct sentence generation but they do not consider the word embedding perspective.

DAD[14] is a fully NAT method that utilizes different attention mechanisms and two additional phases of training for NMT tasks. This approach operates at the word embedding level. It first employs a filtering process to create word embeddings containing relationships with target-side tokens. Once the word embeddings are prepared, DAD utilizes transformers[6] for the subsequent downstream translation task. It uses three training phases to better capture the dependencies on the source side. More details will be discussed in Section 3.2.1. However, Latent variables need training for additional variables, which requires designs for supervising the training. There are masked strategies that are commonly used in NLP such as Bert[7] and only require information from context without additional efforts for deciding the meaning of latent variables carrying or supervision methods.

Masked strategy: We can not only capture information from the relationships between source and target side tokens but also extract relationships between tokens within a single language. The masked strategy[34] proves valuable in comprehending context. This approach trains the model by consecutively masking the decoder input, using an n-gram loss function to mitigate the challenge of translating repeated words. Furthermore, masks can be used to selectively reveal tokens during encoding to gather more information. GLAT[35] involves glancing at the target's ground truth during the initial epochs of training. Consequently, it leverages information from the target sentences, thereby acquiring additional target side dependencies. As the training progresses, the ratio of glancing is gradually reduced to zero, transitioning it into a fully NAT model.

These strategies above focus on the architecture or methods within the model. However, each model needs supervision from loss between ground truth and predictions. There are criteria that could suit the NAT work well.

Criteria: Regarding the loss mentioned above, the traditional Cross-Entropy Loss[36] is not as effective as other criteria. Criteria select different loss functions to guide model training, including CTC loss[37], aligned Cross-Entropy[38], and more. CTC has an algorithm to align the lengths of the predicted and ground truth target side language, resulting in improved loss calculation. Subsequently, gradient descent becomes more manageable to execute.

Pre-trained model: Apart from the above aspect requiring special designs, there are easier ways to leverage a pre-trained model, which can expedite the refinement of the current model. Pre-trained models transfer information (including target side dependency information) to aid the NAT model, as demonstrated in the research by Xiaobo Liang's team[39]. This joint training benefits from the AT model, which retains information within the decoder parameters and offers an initialization point for the NAT decoder.

There are more ways to get help in obtaining information from target side dependency from AT models. The **iteration-based** model strikes a balance between AT and NAT approaches, achieving a trade-off between time efficiency and output performance. This model re-uses the output as a condition for decoding refinement, although it does not achieve the expected speed acceleration. The Levenshtein Transformer[40] introduces deletion and insertion policies to enhance decoder flexibility. It can dynamically adjust the length of target-generated tokens and revise sentences by emulating human actions like undoing, deleting, or replacing words. However, each refinement stage requires an iteration, resulting in time consumption. Other methods have the same time consumption flaw. For example, the masked strategy[41] involves applying masks to tokens with the lowest confidence, initiating iterations to regenerate them using information from other generated target tokens. Consequently, this method gains more target dependencies but necessitates multiple iterations to achieve desirable generation quality. The research reported here focuses on fully NAT instead of iterative decoding since it sacrifices efficiency and would be similar to AT if the iterations are numerous.

The aforementioned solutions primarily address target-side dependency issues without a significant emphasis on the translation aspect. However, there are other challenges in translation tasks stemming from multilingual aspects and the broader field of XMT, which will be thoroughly discussed in the following section.

2.3 XMT, shared embedding and word alignment

In traditional classical XMT tasks, Byte-Pair Encoding (BPE) is a widely chosen method for the encoding process. BPE is a tokenization technique used to segment long words into parts that may reveal some of the meanings within those words, such as prefixes or suffixes. By employing this approach, BPE tokens become shorter and shared, thereby better representing the word's meaning within the embedding space. For instance, the words *art* and *artist* would both share the BPE token *art@@*, with *artist* related to the meaning of *art*, and *artist* can be considered a derivative form of *art*. Utilizing BPE tokenization conserves memory by reducing the size of the dictionary and provides cues to the embedding space to cluster words. Many NMT models, including DAD, adopt this tokenization method, whether they use separate or shared embedding spaces.

In contrast to conventional NMT models, DAD maps both the source and target languages onto a joint embedding space, enabling the model to readily identify

words conveying identical meanings across different languages. This joint embedding space, also referred to as a shared embedding space, falls within the category of cross-lingual embedding spaces. Traditional cross-lingual embeddings employ separate spaces for distinct languages and establish mappings between them. However, Aitor's team[42] highlighted the challenges associated with word alignment in this approach.

Cross-lingual Pre-training[43] adds a language label as a condition, in addition to token embedding and position embedding. It largely alleviates English-centric bias by training the word in a shared embedding space cross-lingually. Due to the effects of shared or joint embedding space, the model built in this research uses DAD joint embedding space. However, this joint embedding space has one severe issue[44] which is that the words tend to cluster according to language, with the language label as a signal, while the shared embedding space should represent the semantic similarity of words.

There is previous work on improving cross-lingual representation. Instead of comparing single words, INFOXLM[45] uses contrastive learning, encouraging sentences that express similar meanings to cluster and convey opposite meanings to disperse. XLM-K[46] designs another approach to align the relations between different languages. They built a knowledge graph with entities from the multi-lingual corpus and linked shared related entities as multi-lingual knowledge. By object entailment, the knowledge graph will be more complete and contribute to the XLM model as a pre-training model. However, it requires a large size corpus and many pre-trained tasks. However, these works are comprehensive and not easy to be applied with other models. Also, it requires additional data such as knowledge graph construction.

CAMLM model

Ernie-M[17] tackled the previous problem by using CAMLM efficiently. It would only refer to information from different language tokens when training certain language tokens. The attention matrix would be partially masked for the tokens in the same language while the attention scores are approachable from the other language. This would force the model to learn the semantic meaning more with less distraction from the language itself. This CAMLM-like method would help to solve the gap that DAD has. More details are illustrated in Section 3.2.2.

Word alignment As the CAMLM component primarily addresses the word clustering issue to improve word alignment, there has been previous work dedicated to word alignment strategies. Lucia's team [47] proposed three methods: one-on-one alignment, one-language-on-all-the-other-tokens alignment, and a combination of both for multilingual translation. The one-to-one alignment, depicted in Fig. 2.3 (a), is particularly relevant to the research conducted in this work, whereas the (b) and (c) are the latter two strategies.

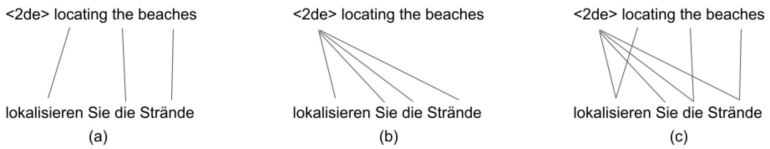

Figure 2.3: One2one alignment[47]

It provides a supervised learning approach for explicit word alignment; however, it requires an external word alignment dictionary. Statistical methods are mentioned in Franz J. O's team's work [48]. The Dice coefficient [49] is selected to measure the aligned word pairs, as shown in Equation 2.3:

$$\mathrm{dice}(i,j) = \frac{2 \cdot C\left(e_i, f_j\right)}{C\left(e_i\right) \cdot C\left(f_j\right)} \text{[48]}$$

where e and f represent two sentences and $C(e_i, f_j)$ means the co-occurrence of count of two words e_i and f_j. Subsequently, by utilizing the largest Dice coefficient, the word alignment can be obtained as a pre-trained vocabulary dictionary. There are existing libraries for word alignment such as GIZA++[1]

[1]https://www2.statmt.org/moses/giza/GIZA++.html

Chapter 3

Methodology

3.1 Issues to address

In the previous section, this work briefly introduced NAT and its related previous work with gaps to be filled. In this chapter, the work builds upon this and introduces the NAT problems to be solved in XMT translation task. Building upon the DAD model while applying NAT model in XML field, there are still two outstanding issues that need to be resolved. Solving the two issues would mitigate the previous two problem to some extent at the same time.

Word Representation

In a joint embedding space, and the traditional approach of word embedding training in XMT, words tend to cluster together based on language more than on semantic meaning. For instance, words like *cat* and *dog* might cluster in the ”animal” section, but *cat* and *Le chat* could be distant from each other as they are not sufficiently similar from a language perspective. This could impact the performance of the DAD because when it establishes relationships between source and target tokens exclusively, aligning words might become challenging due to information leakage from tokens within the same language. The CAMLM-like module in my research model tries to solve this issue because it could eliminate the effects of language and emphasize the role of the meaning of words. (More discussed in Section 3.2.3)

Same character sequences in different languages

The same character sequences share embeddings in the joint embedding space, even though they might differ significantly across different languages. This could confuse the model during its processing of such sequences. While the DAD approach might provide some mitigation, it could complicate the filtering process. Sequences that are present in the source language need to be filtered out, but if they reappear in the target language, they should be reintroduced. LBPE would distinguish the same character sequences according to their language perspective which is a language prompting.

3.2 Model Design: Word-alignment emphasized Dependency Aware Decoder (WDAD)

3.2.1 DAD introduction

The issue of separating points in the shared embedding space arises from the DAD methods themselves. It is further mathematically demonstrated here, building upon the illustration in Fig3.1.

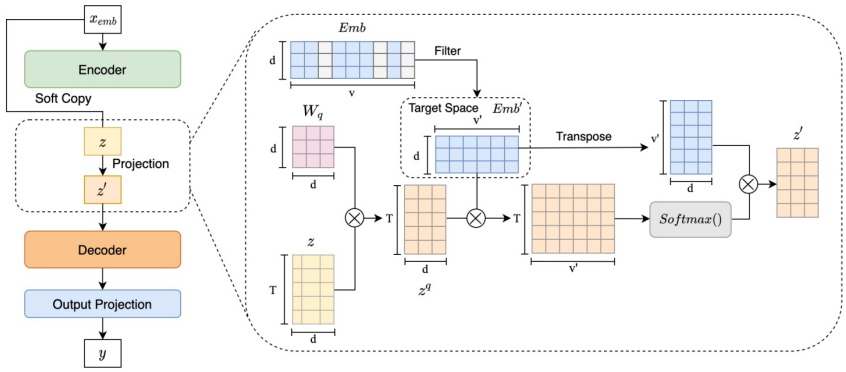

Figure 3.1: DAD attention[14]

Filtering process

Before the core transformers-like module for translating, DAD applies an attention mechanism between the output of the encoder and target tokens embedding to capture the semantic similarity with a filtered dictionary. The filtered dictionary aims to keep only target tokens embedding to avoid information leakage from source tokens in the attention mechanism. It only finds relations between the output of the encoder and the target side tokens regardless of the source side tokens' information leakage.

Mathematically, DAD[14] provides details of the attention mechanism and is shown in Fig 3.1. It selects either the output of the encoder or copies the original embedding of the tokens X as variable z. Then a filtered dictionary is created with only the target token index inside, which filters the embedding matrix and leaves only the target token column. This could pose an issue as shared character sequences may originate from both source and target tokens, leading to confusion. Evidence shows that out of a 39k-sized dictionary in En-De WMT14 translation dataset, only 4k tokens were filtered out. This indicates the effects of the filtering process are not brought out the most effectively because many shared character sequences cannot be filtered out. Solving this problem would contribute to the goals of this work. After filtering unrelated words, it sets the query generated from z and key and value in the attention-like mechanism generated from the filtered embedding matrix to realize the attention mechanism to get z'. z' therefore would contain information

from the source side and the target tokens´ information. More specifically, it could be formulated as follows:

$$z' = softmax(W_q \cdot z \cdot Emb') \cdot Emb'^T$$

This filtering process, together with the attention mechanism, emphasizes the connections between target side tokens and source side information during encoding within the shared embedding space to achieve improved alignment. The attention mechanism attempts to provide an initial decoder state, in target embedding space, based on the similarity of source words to target words. As a result, the information from the encoder becomes more focused on target side details during decoding.

Three phase training

After the process of attention mechanism with the filtered dictionary, in other words, the preparation for the input for the encoder, DAD would experience three phases of training. The first two would be *at-forward* and *at-backward* which aim to extract information from previous and later tokens and is the preparation for the last phase. Then the final phase *NAT* will be bi-directional. The DAD training phase is shown in Fig3.4

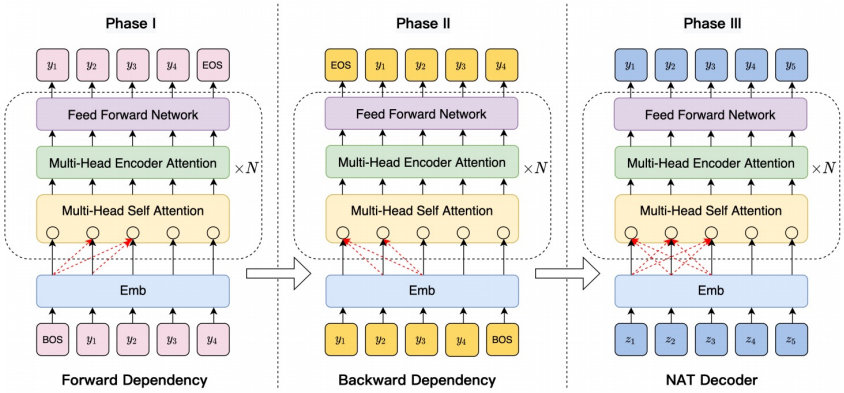

Figure 3.2: DAD three phase training[14]

The diagram illustrates that the *at-forward* phase employs a training strategy, such that when calculating attention scores, tokens can only refer to the previous tokens. In contrast, in the *at-backward* phase, tokens can only refer to the attention scores of the subsequent tokens. These methods emphasize information extraction from one side of the sentence, resembling the role of the pre-training stage. The final *nat* training phase combines the two, allowing tokens to reference both preceding and succeeding tokens, which aligns with the traditional transformer strategy.

Flaws

These two innovative strategies, the filtering process and three phases of training, are introduced within shared embedding spaces, where tokens from both source

and target languages are comparable, providing opportunities for calculating cross-lingual similarity scores. This enhancement improves alignment. However, though the attention mechanism aligns target tokens with source sentence semantic meaning, it does not bring together words with similar meanings in the embedding space. It can be visualized in Fig3.3 below.

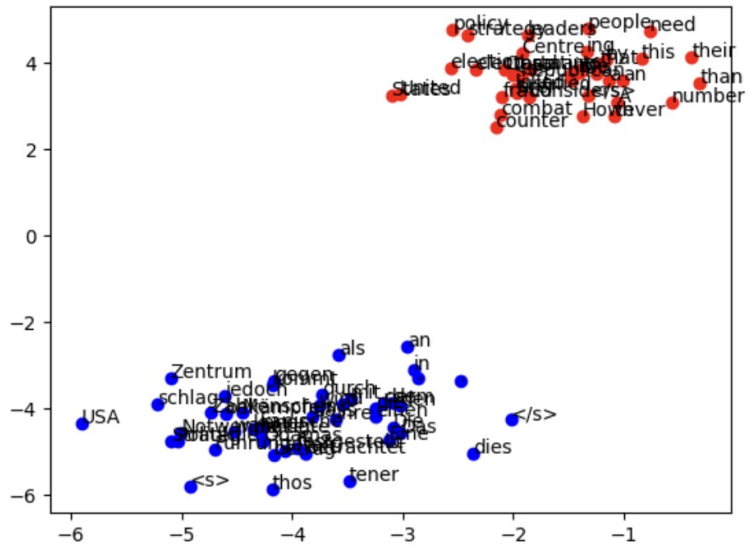

Figure 3.3: Vallina DAD

These are the directions for improvement on word embedding. New versions of word embedding could provide indications of which words have been decoded, thereby mitigating issues of over- or under-translation. This challenge represents a critical domain in XMT, shared embedding spaces, and NAT.

3.2.2 Shared embedding space

The key strategy in this work revolves around employing a shared embedding space instead of separate embedding spaces for individual languages. It's important to note that the filtering process operates exclusively within the shared embedding space, leveraging a unified vocabulary size to eliminate unrelated tokens. There are several other reasons for opting for a shared embedding space in this work.

Firstly, a shared embedding space facilitates linking equivalent tokens, enabling more accurate alignment. Tokens share the same embedding space, ensuring uniform semantic representation across languages, and aiding in capturing cross-lingual relations. While separate embedding spaces could achieve alignment through cross-attention mechanisms from the encoder side, the lack of integration between different languages' embeddings is a limitation.

Secondly, while separate language embedding spaces preserve language-specific features due to tokens originating solely from a single language, this presents a challenge to address. The objective is for the embedding space to convey semantic meaning while minimizing the influence of language-specific information.

Third, utilizing a shared embedding space results in a reduction in the number of parameters, which is less than doubling the vocabulary size required for a separate embedding space. This represents a trade-off between saving the number of parameters and sacrificing the ability to represent words distinctly.

In the context of DAD, the full potential of these advantages remains underutilized, as the traditional encoding methods employed lack modules emphasizing cross-lingual relations. The subsequent section will delve into related models to address this issue.

3.2.3 CAMLM model

The issue of the cross-lingual shared embedding space can be alleviated by the use of a cross-lingual model. Cross-lingual embedding aims to cluster word embeddings that are more closely related in terms of semantic meaning, making word alignment easier. The CAMLM model is one of the cross-lingual models designed to facilitate the learning of semantic information between different languages. Building upon the introduction in Section 2.5, a more detailed explanation of CAMLM is presented in Fig3.4.

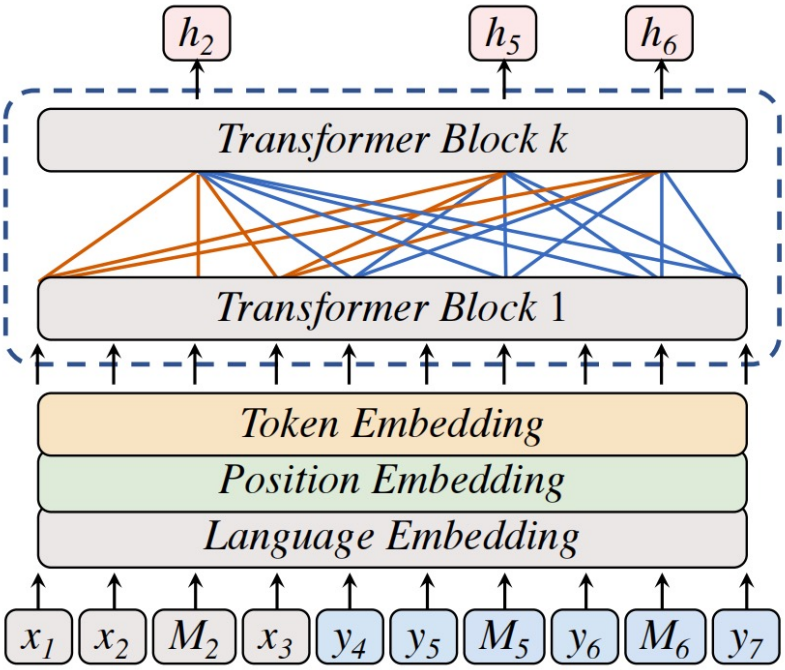

Figure 3.4: CAMLM[17]

The x_i and y_i represent tokens from two different languages. The language pair

$[X, Y]$ with masks is input in the CAMLM model. The M_2, which is the mask of the language of X can only refer to the other language tokens y_4, y_5, y_6, y_7, and the same is as M_5, M_6. The model would need to predict the tokens of the mask and train the word embedding. Feeding those into embedding layers and transformer layers, the model would predict the masked tokens. However, the self-attention mechanism in the transformer of CAMLM model is slightly different. This could force the model to focus on the semantic similarity between different languages. In other words, the tokens would focus more on multi-lingual semantic relations to cluster based on word meaning but not language.

The attention matrix is depicted in Fig3.5, where the dark blue colour represents attention scores utilized in multi-head self-attention, while the light blue colour indicates attention mechanisms that are blocked, which is similar to the concept of masked attention. For example, M_2 could only get information from tokens in Y, therefore, the cross attention scores between M_2 and X are coloured light blue while the score of (M_2, Y) pairs are coloured dark blue.

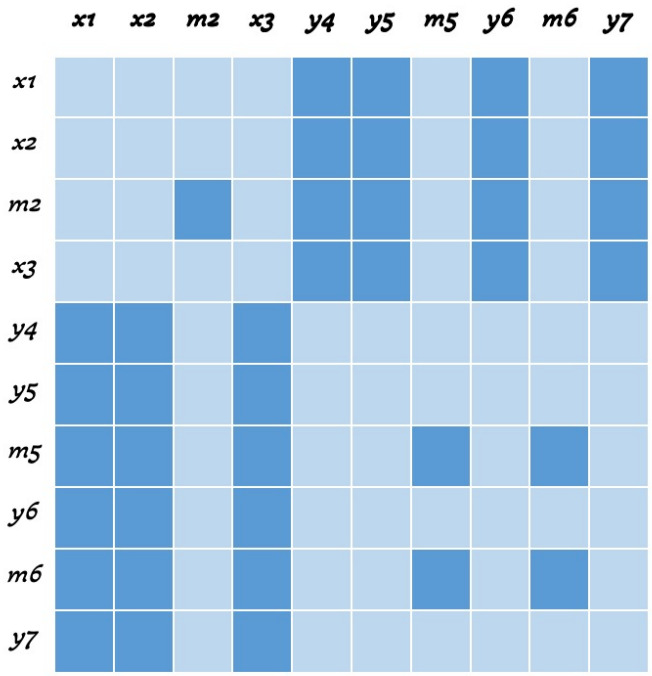

Figure 3.5: CAMLM attention

The model employs a specialized masked attention strategy to achieve the goal of word alignment rather than language-based clustering because it prevents the words from acquiring information from the tokens in the same language. This could be shown in Fig3.6. The blue points represent German tokens, while the red points represent English tokens. It can be observed that the words cluster closer regardless of their language.

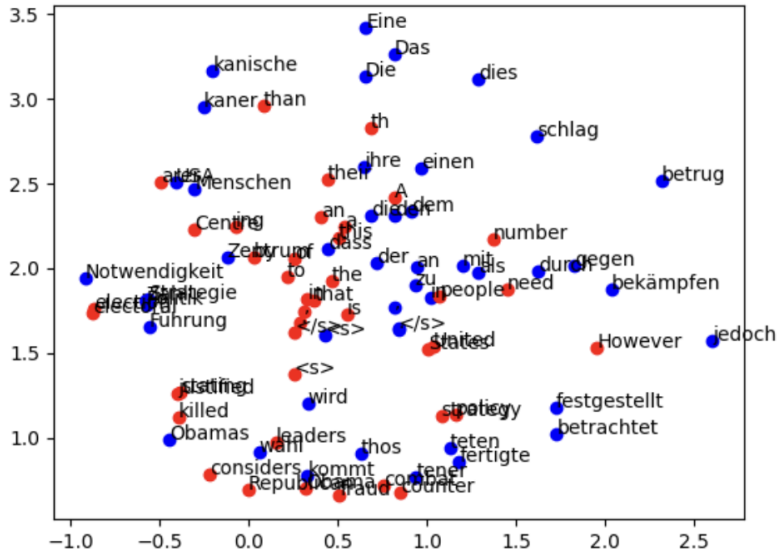

Figure 3.6: CAMLM pure

3.2.4 Word alignment supervision

However, word alignment is compromised by the subsequent DAD training strategy because the traditional DAD training has the source tokens from the same language, unintentionally leads the word reclustered according to language. The model therefore introduces explicit word alignment supervision during DAD training to preserve the word alignment features. As discussed in Section 2.3, the approach initially establishes a pre-trained word alignment dictionary using the Dice coefficient. Subsequently, the model employs this dictionary to generate new word-aligned sentences C token by token. Following this, the model incorporates an additional loss term between the predicted tokens and the word-aligned sentences, providing explicit supervision, as illustrated in Equation 3.2.4.

$$Loss = Crossentropy(A, B) + Crossentropy(A, C)$$

where A is the predicted output and B is the reference. When there is additional term emphasizing the word alignment during training, the pre-trained word embedding is predictably not badly influenced by traditional DAD training.

The word embedding issue cannot be solved purely on DAD because there are many shared character sequences and its filtering process cannot distinguish them. The different BPE tokens from the two languages are few (4400 unique En tokens out of 39984 tokens for De-En translation task in the Ninth Workshop on the WMT14 De-En dataset) which would impair the effects of the attention mechanism. Also, DAD would not be forced to learn relationships between tokens from different languages. With the flaws of DAD, distinguishing between shared character sequences would enhance the filtering process's effectiveness because this way the filter dictionary would only include tokens from the target language. The next subsection explains the detailed approach for dealing with the issue.

3.2.5 Data augmentation: LBPE

For traditional word tokenization, BPE[18] tokens are effective in restructuring the vocabulary, transitioning from single characters to sequences within a single language. The principle of how BPE tokens are generated is demonstrated in three steps. First, the initial dictionary consists of single characters. Then, it counts the frequency of combinations of characters in the dictionary and selects the most frequent ones to form new tokens in the dictionary. This process is repeated until it reaches the stopping criteria.

However, the advantage of sharing sequences within a monolingual context proves to be a drawback in a multilingual embedding space. This is because the shared sequences may not convey the same meaning across different languages. Therefore, this research introduces language-based BPE tokens (LBPE) to differentiate the shared sequences originating from different languages as a method of data augmentation strategy.

Inspired by Shon's research on language embedding [50], this project revised the traditional BPE approach by appending language-specific indicators after BPE tokens, as illustrated in Fig3.7. In other words, the BPE tokens would be revised with language suffixes. In this representation, *BPE*@@ represents the original BPE tokens, and *lan* serves as the language signal. For example, *cat*@@ would be transformed into *cat*@@*en*. (It's worth noting that in experimentation, for implementation convenience, it might appear as *cat*@@_@*1*@_ , representing the source or target language. Another reason for choosing this format is that, while some BPE tokens could end with _ , very few tokens will contain both @ and _ simultaneously.)

Figure 3.7: LBPE

This has a significant impact on the dictionary. For instance, the original BPE dictionary of the WMT14 De-En dataset has a size of 39,840, while the LBPE dictionary has a size of 70,968. This difference indicates that there are 31,128 identical BPE tokens being shared. Upon applying LBPE, there will be no identical character sequences except for certain signals like commas. Each character sequence can have a distinct representation in the embedding space, thereby eliminating confusion across different languages. At the same time, the number of vocabulary size is increased thereby increasing the representation capability of the model. Additionally, this approach redefined the tokenizer, which is the format of data augmentation.

3.2.6 Combination of CAMLM and LBPE model

This research incorporates two modifications to DAD, which are individually applied first and then combined. Regarding LBPE, the focus is on the dataset itself, resulting

in changes to the dataset and its corresponding dictionary according to the rule described earlier. As for the CAMLM-like component, its primary objective is to enhance word embeddings. Consequently, integration should occur before the main training phases.

As mentioned previously, CAMLM from the Ernie-M model applies distinct attention strategies to prevent information leakage from a single language. Following the application of the attention strategy, token embeddings are enhanced by incorporating similarities between pairs of languages. Hence, this research adopts techniques akin to pre-trained models to replace the word embeddings within the DAD model, which is the design of the combination model, which utilizes the language of the tokens as a prompting indicator. The approach is shown in Fig3.8.

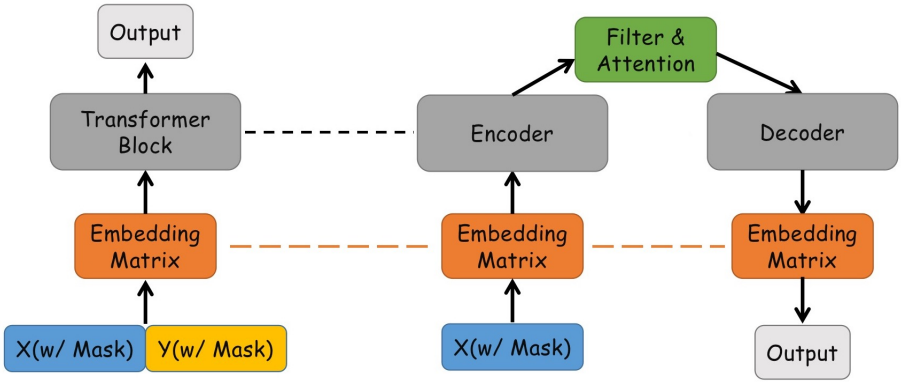

Figure 3.8: Combination model architecture

With the size of X, Y concatenated, the language pair with masks is fed into the Embedding Matrix, and the transformer block undergoes training. The left portion resembles a CAMLM-like network. It needs to be trained prior to commencing the main combined model training. Subsequently, the DAD network is trained, utilizing the same Embedding Matrix as the left network, and the Encoder is initialized with relevant parameters from the transformer block of the left network. In addition, the encoder would be regarded as a whole component to hold the information of the pre-trained model instead of just one embedding layer. Therefore, the whole encoder, including multi-head self-attention, is also initialized with relevant parameters from the transformer block. This enables the DAD network to draw benefits from the pre-trained left network. DAD now can have word embeddings without information leakage from other languages, which potentially enhances accuracy and reduces training time.

The aforementioned strategy aims to partially alleviate target-side dependency issues through improved word embeddings that differentiate words across languages. However, the results from running the model indicate that the combined approach falls short compared to individual methods, especially the LBPE strategy (further details discussed in Chapter 4). This limitation might be attributed to LBPE's introduction of a larger vocabulary size, which complicates the training of the CAMLM-like

module because it provides more parameters to be trained. Additionally, conventional translation strategies typically involve considering only one language on the source side, leading the CAMLM-like encoder to implicitly learn language-specific effects. In essence, since all tokens inputted into the encoder originate from a single language, the encoder might inadvertently treat language as a latent feature, thereby influencing the clustering of word embeddings not solely based on semantics, but also due to language-related factors. Therefore, the next subsection introduces a data-mixing strategy and language embedding layer to mitigate these issues.

3.3 Language embedding layer and Mixed data

LBPE can distinguish the tokens with their language sign for the same characters but have a large vocab size for more parameter training to verify the effects of language promption data augmentation strategy. This research model aims to maintain the idea of language signs but reduce the number of parameters and vocab size. Therefore, as the embedding layer in the encoder extracts different aspects of features, this work designs a language embedding layer on top of the current embedding layers. The input is the language sign and the output is the hidden state with the same dimension as the other embedding hidden states, shown in Eq3.1. Then, similar to the position embedding, the language embedding is added to the hidden state of embedding, shown in Eq3.2.

$$Emb_{language} = f(sign) \tag{3.1}$$

$$h = h(other) + h(Emb_{language}) \tag{3.2}$$

This method is expected to reduce the vocab size but retain the differences if they are from a different language because it allows the same character sequence to create shared token embedding but with different language embedding. However, there is one issue that still exists, since the source side data will only be from one single language, which leads to the input of the language embedding layer remaining the same, and then influencing the effect of the CAMLM-like module.

To solve this issue, this work proposes the mixed data strategy. Instead of translating from one language to another language, the data set would be shuffled more multilingually. There would be language from both the source side and the target side. For example, for the De-En translation task, there would be English and German as sources and targets at the same time. This way, the encoder would have multi-lingual input and the effects of the language from encoding would be mitigated. Also, the input of the language embedding layer would not stay the same.

3.4 WDAD

As stated above, multiple methods have been proposed. Some aim to address word alignment issues, while others focus on shared character sequences or dataset-related issues. WDAD is a combination of the best-performing models from each component.

The base architecture will still remain the same as DAD, but there are choices regarding how to supervise word alignment and the data augmentation strategy. Moreover, choices from the two aspect of improvements will not contradict each other for the changes made in architecture.

Chapter 4

Evaluation and Experiments

4.1 Datasets

The experiments include reproducing the DAD result on the benchmark WMT14 De-En dataset and also evaluating on the multi30k data set and WMT16 En-Ro dataset.

The Ninth Workshop on Machine Translation (WMT) is widely used in NMT task experiments and involves many language pairs including English, French, German, Russian etc. These are mainly taken from [51] and used to check the performance of NMT models. It has approximately 4.5M sentences in De-En task. Multi30k is another smaller data set including language pairs to examine the NMT model. It has approximately 30,000 sentences in De-En task[52].

This research mainly focused on translating from German to English (De-En) task. It has 4.5M pairs of language, for which this report applies the same prepossessing of the DAD paper[14]. The NAT models in this experiment are trained on distilled data, which is also the same as DAD[14]. For implementation evaluation, multi30k is selected and this research used the pre-processing step on Fairseq[53].

Additionally, to assess the scalability of the research model, an additional dataset was incorporated from WMT16, akin to the one used in WMT14. Specifically, we focused on the En-Ro dataset within this collection, which comprises 608,319 examples—slightly smaller in size compared to the De-En dataset.

4.2 Comparative approaches

This research contains two primary solution ideas. In order to assess the individual contributions of each component, and the combined impact, four experiments were designed: LBPE alone, CAMLM replacement alone, the combined use of LBPE and CAMLM, and the baseline Vanilla DAD. In other words, for LBPE, this report utilizes a distinct dictionary with a new tokenizer, while for CAMLM, it substitutes the encoder of DAD. For the combination methods, both the tokenizer and encoder are replaced simultaneously. Additionally, this work conducts additional training

experiments focusing solely on the third phase of DAD. This approach is based on the assumption that three-phase training might be overly potent, overshadowing the CAMLM training. The basic analysis is mainly focused on the WMT De-En dataset.

Given the limitations of combined work discussed earlier, this work conducts additional comparative experiments to evaluate the effectiveness of the newly designed strategies, namely CAMLM and CAMLM + language embedding, when applied to DAD. The goal is to assess whether these strategies mitigate the identified limitations. Furthermore, experiments involving CAMLM + word alignment supervision are conducted to investigate whether word alignment supervision addresses the challenges associated with word embedding(generated by CAMLM) impairment caused by DAD. There are additional comparative experiments in Section 4.6 for sanity check about the effects of CAMLM and data augmentation strategy.

4.3 Hyperparameters

To maintain parallelism with the DAD paper, all hyperparameters within the DAD model remain unchanged, which is also consistent with the research by Qian's team [35]. For both datasets, the base-Transformer was employed, with a hidden dimension of 512, 8 multi-heads, and 6 encoder and decoder layers.

Additionally, in this experiment, the learning rate and early stopping patience number are fine-tuned as they significantly influence convergence speed and the potential for overfitting. The patience number signifies the number of epochs during which the validation set score fails to exceed the best score, after which training halts. Given the substantial size of the dataset, training a single epoch of multi30k takes approximately 5 minutes, while WMT14 De-En requires around 1 hour and 20 minutes, even with the utilization of 5 GPUs (GeForce RTX 2080 Ti Rev. A) in parallel. Extending the patience duration would result in increased computation time while a shorter patience duration could result in underfitting. Ultimately, this report employs a patience value of 4 for WMT14 and 10 for multi30k, while the learning rate is set to 1e-4.

Additionally, the parameter max_tokens is also adjusted to accommodate GPU limitations and gradient descent considerations. A larger max_tokens value accelerates training by truncating and grouping sentences more effectively, thereby allowing weight backpropagation to be calculated on a larger amount of data. This advantage resembles that of batch gradient descent. Finally, this research adopts 2048 max_tokens since the memory of the GPU is not enough for more tokens. However, this would impair the final performances of the model because the texts in the test set exceeding the max_tokens would be truncated. It is shown that for short texts in a validation set of WMT14, the blue score is three times that of long texts in the test set. This might be one reason why the experiments are not up to the score reported in Jiaao et al.'s research[14].

Importantly, this experiment compares the efficacy of using a network to learn positional embeddings against directly utilizing sinusoidal positional encoding, as out-

lined in Transformers[6]. The former yields approximately +1.5 BLEU score increase compared to the latter. This finding suggests that learnable positional embeddings require much training time but offer a better fit for the dataset itself.

4.4 Results

Following the fine-tuning of all hyperparameters, this report initially presents the results for multi30k, which are displayed in Table 4.1. For this experiment, five repeated tests were conducted using different seeds, and the average is reported. The ”DAD trained for NAT phase” section refers to testing without the *at-forward* and *at-backward* phases, only using the *nat* phase for comparison between the 3-phase training experiment.

The following paragraphs mainly analyze the effects of CAMLM and the data augmentation strategy LBPE with theoretical support. The dataset Multi30k serves as the sanity check and debugging dataset, while the formal analysis is conducted on the WMT14 and WMT16 datasets. After elucidating the effects of these components and the combined model, the subsequent paragraphs delve into additional models, including word-alignment supervision, language embedding & mixed data strategy. And more other comparative experiments are explained in the next few sections.

The reason for using this ablation study experiment is that the last phase of DAD combines the training strategies of the first two phases, which involve acquiring information from either previous or subsequent tokens. In other words, the last phase represents an enhanced training strategy composed of the first two phases. This also explains why the work assumes that the first two phases are redundant. Also, training with a similar strategy repeatedly would be assumed to diminish the effectiveness of CAMLM-like components because it is the pre-trained part, and has low capability to maintain its information in another training strategy. However, the results confirm the assumption that the effects of CAMLM components will be diminished by the initial training stages was incorrect. This can be attributed to the fact that the CAMLM component primarily concentrates on improving word embeddings, whereas the first two phases prioritize gathering content-related information. These two aspects of information acquisition could differ significantly even though the three-phase training has proved powerful enough. Hence, in order to save computational time, this research did not conduct experiments on ”DAD trained for the nat phase” for WMT14.

Table 4.1: Multi30k result

Model	BLEU score
Vallina DAD	32.234
LBPE DAD	**32.366**
CAMLM DAD	31.17
Combined model	31.372

The LBPE works slightly better while CAMLM performs slightly worse than the baseline for multi30k. This discrepancy can be attributed to the simplicity of the multi30k dataset. Due to the brevity of sentences and the small vocabulary size, even if the word embedding space lacks clustering based on semantic meaning, the translations can still make the alignment. Therefore, this research demonstrates it is worth investigating for WMT14 De-En experiments.

For WMT14 De-En, this report conducts three repeated tests and reports the average results, as shown in Table 4.2.

Table 4.2: WMT14 De-En result

Model	BLEU score
Vallina DAD	19.213
LBPE DAD	**19.84**
CAMLM DAD	19.613
Combined model	19.31

The results indicate that LBPE proves effective for both datasets, while CAMLM's effectiveness varies. Moreover, the combined components of the two revisions also demonstrate effectiveness. (Note that this research does not achieve the scores presented in DAD[14], which could be due to variations in hyperparameters or differences in hardware configurations.)

As depicted in Table 4.1 and Table 4.2, the BLEU score for the LBPE experiments on both datasets increases, even achieving a +0.6 BLEU score improvement for WMT14 De-En. This enhancement might be attributed to the substantial number of shared word tokens. This is the shared sequence problem mentioned in Section 3.1 that could potentially confuse the model's ability to adjust token embeddings accurately. This could be verified by cleaning out all samples containing shared character sequences and comparing the model's performances. LBPE addresses this issue by assigning distinct embeddings to tokens from different languages, allowing them to be separately trained while maintaining consistency in terms of content usage. For example, *ant@@_en_* in *antibody* and *ant@@_de_* in *enseignant* would be trained independently but not in the original way that the token embedding of *ant* would lead to two different word sections in embedding space. This may understandably become more accurate when the language shares more words that carry distinct meanings because now the words in different languages can only have one direction to learn. English and French exhibit similarities, resulting in numerous shared words that express identical meanings, like *justice* or *counsel* and *conseil*, thereby leading to redundancy at this stage.

The influence of the CAMLM component is small on the final BLEU score performance for WMT14 De-En, which is +0.4 BLEU points. This is because CAMLM is preventing information leakage and CAMLM's ability to enhance word embedding clustering is a positive indication of progress (further elaborated in the following section). The enhancement of word embeddings could yield even greater effective-

ness when dealing with more than three languages, as this would introduce more complexity.

To integrate both solutions, this research conducted experiments using combined model. It outperforms the baseline BLEU score (+0.1) slightly, although it doesn't achieve the same level as the two methods acting independently. This could potentially be attributed to the fact that LBPE increases the vocabulary size, while CAMLM reduces the training content by rendering some identical character sequences distinct. The later experiments on CAMLM with a language embedding layer would test whether reducing the vocab size could mitigate this. One conceivable approach might involve discarding LBPE and instead utilizing language embeddings to differentiate tokens from different languages. By doing this, more corpus can be used to train individual character sequences compared to the cases where shared sequences are distinguished by LBPE when fewer corpora are applied to train the tokens, and simultaneously, language embeddings could be fine-tuned to better suit the data.

Therefore, this work reports the experiments of the language embedding layer. This would keep the advantages of both CAMLM and LBPE but with fewer parameters to train. Also, the mixed data strategy is propsed to address the issue that DAD might impair CAMLM has a method proposed, and there are experiments on this. This work performs experiments on the WMT14 De-En pair and WMT16 En-Ro pair, with the results shown in Table4.3

Table 4.3: BLEU Scores for Different Methods on More Data Pairs

Data pair	**Method**	**BLEU score**
WMT14 En-De	Vallina DAD	17.45
	CAMLM	17.64
WMT14 mixed data	Vallina DAD	16.2
	CAMLM	**18.12**
	CAMLM + Language Embedding	16.66
WMT16 En-Ro	Vallina DAD	20.5
	CAMLM	21.2
WMT16 Ro-En	Vallina DAD	21.15
	CAMLM	22.12
WMT 16 mixed data	Vallina DAD	20.45
	CAMLM	**21.46**
	CAMLM + Language Embedding	20.64

The table above shows that for more data pairs in different datasets, which resonates with the results on WMT14 De-En and shows the effects of the mixed data strategy and language embedding layer. CAMLM gives quite good performances. Notably, for the WMT16 dataset, there is an obvious score increase with up to **+1** BLEU scored. This might be because WMT16 does not have too many long texts that are truncated,

compared to WMT14. Also, there are smaller discrepancies between the BLEU score of validation and test set in WMT16 (approximately less than 1 bleu score) than that of WMT14 (around 39 vs 19 in validation and test respectively), which might reveal the test set in WMT14 are not clean enough.

Notably, the combination of CAMLM and the Language embedding layer has slight improvements over the baseline but is a bit worse than the CAMLM on its own. This is potentially because the effects of CAMLM and the language embedding layer are opposite. As mentioned above, CAMLM aims to avoid information leakage, while the language embedding layer distinguishes the tokens from different languages but it makes the model implicitly learn the information from one side. This is because the tokens in one sentence would have the same input language codes and encoding the language codes implicitly emphasizes the information from one side. This reduces the vocab size but still has space for further improvement.

To address the challenge of CAMLM pre-trained embeddings being affected by subsequent DAD training, this research introduces explicit word alignment supervision during DAD training by modifying the loss function (as discussed in Section 3.2.4). The study selects the WMT16 ro-en dataset for experimentation and obtains an external word alignment dictionary by applying Dice's coefficient with the FastAlign toolbox[1]. A portion of the dictionary is illustrated in Fig4.1.

[1]https://github.com/clab/fast$_{a}lign.git$

```
"rezolvarea": "resolve",
"uneia": "of",
"dintre": "the",
"problemele": "issues",
"care": "what",
"contribuit": "helped",
"la": "to",
"criza": "crisis",
"astfel": "so",
"perspective": "prospects",
"stabilire": "establishment",
"piata": "market",
"increderii": "confidence",
"diferita": "from",
"legislatia": "law",
"ilor": "the",
"individuale": "individual",
"provinci@@": "provinces",
"este": "is",
"suplimentar": "supplementary",
```


Figure 4.1: Word alignment dictionary

The alignment of the ro-en language pair appears to be robust based on the graph. However, there are instances of minor errors, such as *care* being aligned to *what*, possibly due to ambiguous contexts in certain training sentences. Leveraging this dictionary, the study generates aligned sentences for supervision. The training is currently in progress, and the results will be available till the time of the presentation.

Returning to the issues of over/under translation and the multi-modal problem that NAT has, the following section will analyse the over-translation challenge by comparing the four base experiments on the over-translation challenge, either LBPE or CAMLM help mitigate over-translation to some extent, resulting in a reduction in the number of consecutive outputs, as illustrated in Fig4.2 and the multi-modal problem is discussed in Section 4.5.3. The mitigation of over-translation is because the language signals prompt the DAD to give greater consideration to the semantic meaning of words.

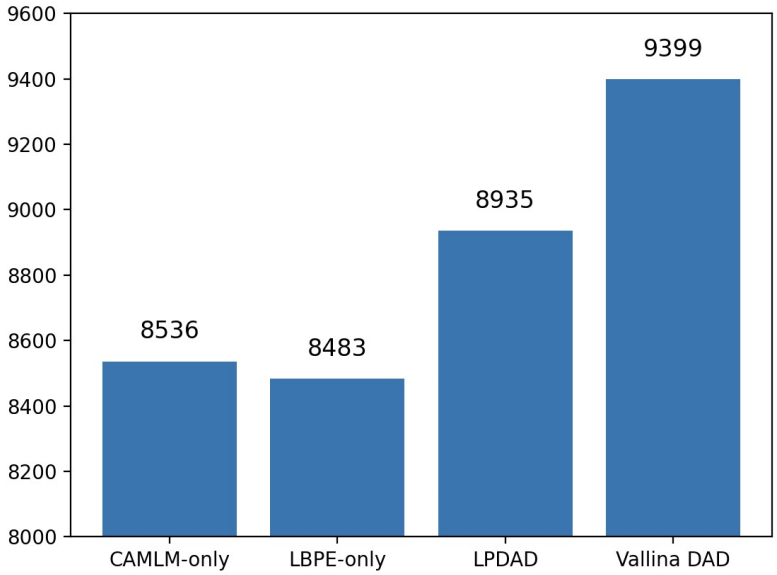

Figure 4.2: Number of consecutive tokens for four methods

As depicted above, the Vanilla DAD applied to the WMT14 dataset produced a total of 9399 consecutively repeated tokens. For example, the phrase *This* ***is is*** *a dog* contributes to 1 consecutive repeated token, representing the existing issue of over-translation. In comparison to the baseline, combined model exhibits a decrease of approximately -400, while the two components functioning independently manifest a reduction of approximately -900. This suggests that the combined model methods effectively address one of the challenges associated with NAT.

4.5 Qualitative Results

4.5.1 Pre-trained CAMLM

The outputs of masked tokens are coherent and sensible. For more rigorous technical verification, the previous plan includes using the Bert model as a baseline comparison. However, due to computational time constraints, the model has not been fine-tuned extensively. Nevertheless, the results from DAD could also serve as supporting evidence, indicating that CAMLM is well-pretrained.

4.5.2 Shared embedding space comparison

To confirm the improvement in word embedding within the joint embedding space, this report employs T-SNE[54] to reduce the embedding dimensions to 2 for better visualization. The graphs presented before (3.3, 4.3, 3.6) compare the embeddings of certain non-BPE token words (those not ending with @@) within the shared embedding space.

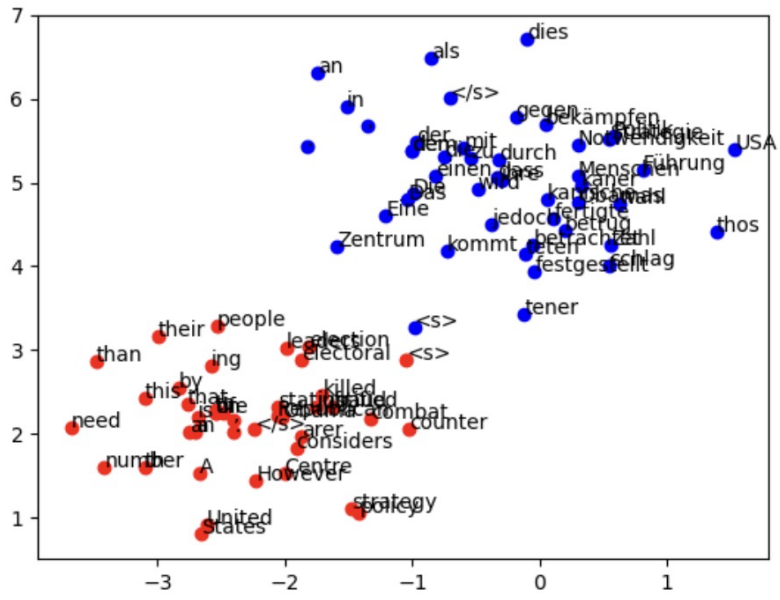

Figure 4.3: CAMLM + DAD

In the figures above, the blue points represent tokens from German (De), while the red points correspond to English (En) tokens. These points are generated by the embedding layers of the best checkpoint model of CAMLM with DAD training. The tokens tend to cluster gradually based on their semantic meanings rather than their languages.

This provides evidence that CAMLM effectively enhances word embeddings, resulting in improvements in the final BLEU score. The reduction in cluster centre distances contributes to more accurate word alignment. However, when CAMLM is integrated with DAD, its performance doesn't match that of standalone CAMLM. This discrepancy could come from the influence of the three robust training phases or the decoding process, which focuses solely on one language. Consequently, this approach might inadvertently impact the word embeddings of the other language. Also, this might be due to the conventional encoding strategy that the translation task has, which is encoding the tokens all from one language. Therefore, this work tries CAMLM with a Language embedding layer on mixed data shown in Fig4.4.

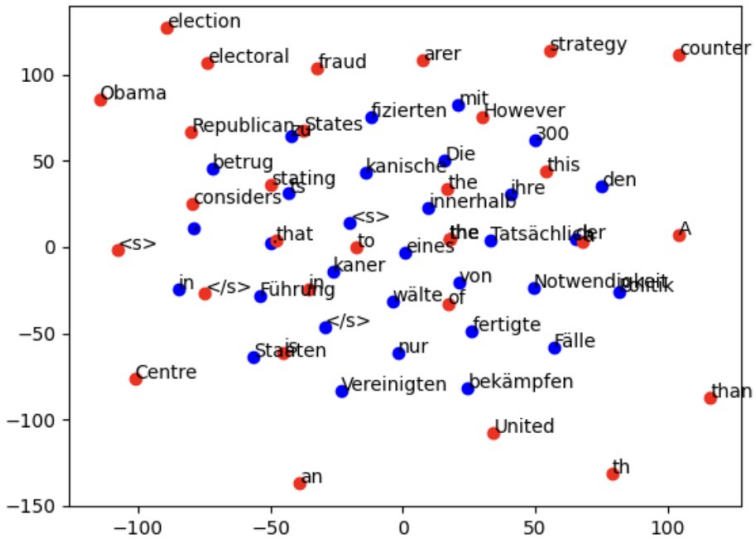

Figure 4.4: CAMLM + Language Embedding on Mixed data

As the figures above show, the blue points represent tokens from German (De), while the red points correspond to English (En) tokens. These points are generated by the embedding layers of the best checkpoint model of CAMLM with Language Embedding on mixed data after DAD training. The flaw of the CAMLM with DAD training, that the tokens fall back to be separate according to language, is mitigated.

4.5.3 Case study

To validate mitigation for the challenges of NAT, namely underfitting and the multitude of translations, this research has selected specific cases for comparison, as presented in Table 4.4.

Table 4.4: Case study

Reference	Children's dreams come true.
Vallina DAD	K's Dremes mes come true
LBPE only	Children's Dreres come True.
CAMLM only	Children's ams Get True
Combined model	Child Dreams come True
Reference	Built by experts.
Vallina DAD	Experts experts.
LBPE only	Designed by ts.
CAMLM only	constructed by experts.
Combined model	Built by experts.
Reference	We must stay composed and keep it up .
Vallina DAD	This is to continue mly mly .
LBPE only	That is to quietly continue .
CAMLM only	This is to to continue calmly .
Combined model	This is to continue quiecontinue .

It can be seen that regarding the three improving methods, it has translated *children* or *child* successfully, which is the under translation problem in Vanilla DAD. Also, the multitude problem *Get true* in example of CAMLM only has been mitigated by other improvement methods. The other two examples demonstrate that CAMLM and LBPE address underfitting and the multi-modal problem.

This case study also highlights a challenge: there are errors in word alignment, particularly involving certain BPE tokens. For instance, the translation of *Dremes* might include *Dre@@* and *mes@@*, whereas *Dreams* consists of *Dre@@* and *ams@@*. Consequently, in the case of CAMLM only, there is only a single word of *ams*, which conveys little meaning.

4.6 Comparative experiments analysis

CAMLM effects together with pre-trained advantages The CAMLM experiments are trained by initializing with pre-trained BERT parameters to expedite training and enhance performance simultaneously. While CAMLM is indeed a pre-trained method contributing to performance improvement, its impact extends beyond mere pre-training; it also leverages the benefits of word alignment power. For comparison, the model trained without applying BERT initialization required 30 additional epochs to converge and experienced a decrease of over 2 points in BLEU score.

The number of parameters of LBPE and shared embedding space There is an increasing number of parameters in LBPE tokenization compared to traditional BPE. Additionally, there is a slight decrease in the shared embedding space compared to

separate embedding space. Although DAD has to apply shared embedding space, it is worth discussing the trade-off balance regarding the number of parameters. To valid the effects of data augmentation LBPE and the influence of increased number of parameter, this work retrained a traditional BPE vocabulary on the distilled dataset WMT14 De-En, resulting in a vocabulary size of 56k in the end. Using the same training strategy, this method yields a 20.99 BLEU score, which is a +2 point improvement over LBPE. This significant enhancement in performance is attributed to data augmentation and the improved representation of word embeddings. These comparative experiments demonstrate the effects of data augmentation. However, they also indicate that LBPE may not be the best method of data augmentation, especially when considering the results of the language embedding layer.

4.7 Sensitivity and limitations

The methods above consistently improve performance compared to the original DAD model. The following graph takes the language pair De-En in WMT14 as an example and illustrates the variations from repeated experiments with different seeds in Fig 4.5. It is evident that, for both CAMLM and LBPE components, significant improvements are observed across all seed values.

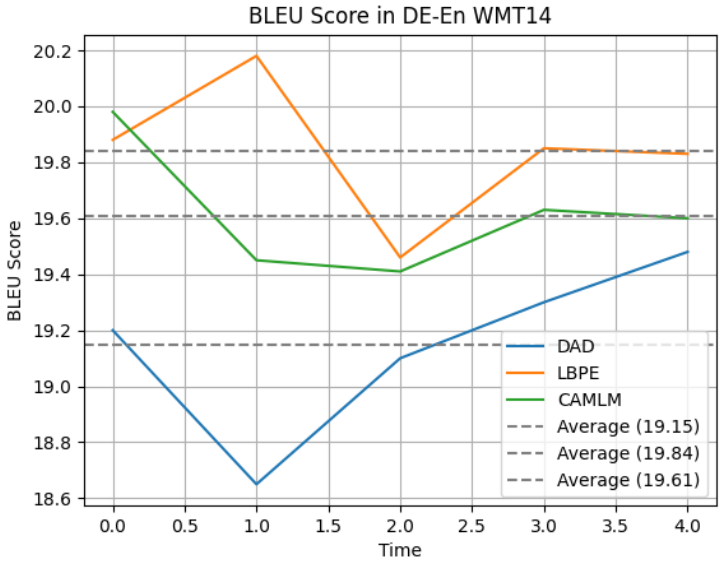

Figure 4.5: Deriviation of experiments

Even though there is an improvement in WDAD methods and its components, there are still some limitations. As shown above, the scores for multi30k are not significantly different, suggesting that the model revision is not sensitive to small datasets. This could be attributed to the complexity of the original model, which is sufficient for this limited-scale data. However, a noticeable increase is observed in the case of WMT14, where the corpus consists of 4.5 million pairs.

Another limitation arises from the complexity of natural language. Some of the reference translations even appear to be less sensible or less straightforward than the model output. For instance, the source sentence *Wasser ist weiterhin kostenlos* is translated as *Water continues to be free*, whereas the reference translation states *It will still give away water*. At times, the reference translations may not be immediately comprehensible or simple enough for non-native speakers. This happens because references in NAT have an accumulation of errors due to downstream erroneous translations of original references. This complexity can make it more challenging for models to grasp the semantic meaning or receive direct supervision. However, it is evident that the model effectively learns word alignment, as the translation remains coherent. Thus, due to the significant divergence between human-crafted speaking habits and model generative strategy, the BLEU score might turn out to be low.

Furthermore, the enhancement of word embeddings relies on the assumption that there exist numerous shared character sequences between languages, like English and French. However, in cases where there are few or no shared character sequences, such as between English and Chinese, the utility of LBPE becomes diminished.

Moreover, this study solely considers language pairs individually. In the field of cross-lingual research, if more than two languages are considered collectively, the dynamics can change considerably. With multiple languages in consideration, a larger corpus could be utilized to reposition word embeddings within the joint embedding space. Additionally, each language introduces its own unique expression habits, leading to potential misunderstandings. Take idiomatic expressions as an example; translating idioms from English to Chinese presents difficulties, and even humans require contextual cues or examples for comprehension. When handling multi-lingual languages, such as translating from Chinese to Japanese and translating Japanese to English idiom, the risk of information loss becomes even more accumulative.

The BPE token itself possesses inherent limitations, which makes it challenging to determine whether tokens have been clustered from a semantic perspective. While the BPE token is a component of the word and carries some level of meaning from the word itself, it is generated based on frequency counts rather than semantic meaning. Consequently, the clustering process might not be as intuitive. For instance, the BPE token *hell* could be part of both *hell* and *hello*, despite their distinct meanings. Moreover, the rules for splitting words to obtain BPE tokens vary for different words, leading to situations where BPE tokens might unexpectedly form components of unrelated words. This aspect could potentially be explored as a future research direction.

Finally, this work compares the performance of the NAT and AT models in each data pair set, shown in Table4.5. It was shown that there are still gaps for accuracy between AT and NAT though NAT would speed up the training process to a large extent.

Table 4.5: NAT vs AT

Data pair	Method	BLEU score
WMT14 De-En	NAT	19.213
	AT	27.72
WMT14 En-De	NAT	17.45
	AT	25.4
WMT16 En-Ro	NAT	20.5
	AT	33.13
WMT16 Ro-En	NAT	21.15
	AT	31.44

Chapter 5

Conclusions and future work

5.1 Conclusion

This work primarily focuses on NAT models within the context of XMT. Upon reviewing existing literature on this subject, the primary challenge observed in NAT pertains to the absence of target-side dependency. This issue manifests in two distinct challenges: over-translation and under-translation, and the multitude problem. While data set distillation can significantly address the multitude problem, the remaining challenges persist in cross-lingual translation. Hence, the DAD model has been formulated to alleviate these concerns, although certain limitations remain to be solved.

Within the context of XMT and NAT, this study aims to enhance performance from two perspectives: improving word embeddings within a shared embedding space and distinguishing shared character sequences from various languages. Drawing inspiration from Ernie-M, this report incorporates CAMLM as a component to be combined with DAD. Additionally, a data augmentation solution named LBPE has been devised to address the challenge of identical character sequences. Consequently, these two concepts form the WDAD model. Also, more methods are proposed to make up for the flaws of each model.

As a result, the BLEU score has improved by approximately 0.5 points. Additionally, word embeddings tend to cluster more according to their semantic meanings. However, the combined model does not perform as well as the two ideas individually. There is still room for future investigation. The table below summarizes the methods employed in this work and provides essential comparisons as the conclusion.

Table 5.1: Strategies Overview

Strategy	**Performance**	**Advantages**	**Disadvantages**
Vanilla DAD	Baseline(0)	Tackling target side dependency to some extend	Word alignment issue & Shared character sequence issue
CAMLM	3	Cluster word only according to semantic meaning	Pre-trained embedding influenced by DAD later training
Data augmentation (LBPE)	2	Tackling shared character sequence issue	vocab size mechanically doubled up
CAMLM + LBPE	2	Combination of above methods	pre-trained embedding worsely influenced & hard to converge for LBPE vocabulary
Data augmentation (others)	4	Better word representation and increased diversity	hard to save training time & fail to solve shared character sequence issue
Word-alignment supervision	To be experimented (expected 4)	Avoiding pre-trained model influenced by DAD later training	requirement of external pre-trained word-aligned dictionary
WDAD (word alignment + data augmentation)	To be experimented (expected 5)	Combination of best performed methods	-

5.2 Future work

More dataset Multi30k is a small-scale dataset and can only validate the correctness of the implementation. It consists of only one set of experiments and one language pair. If more language pairs were considered, the results could be more convincing. There are more language pairs in WMT14 and WMT16, and also famous data set including IWSLT datasets (International Workshop on Spoken Language Translation)[1].

Cross-lingual dataset This model only considers two languages at a time. If more than two languages were considered, there might be more interactions between each pair of languages, and CAMLM could become more effective.

Word alignment experiments & CAMLM components revision The CAMLM model can cluster word embeddings effectively, but its impact is hindered by DAD. The decrease in performance may be attributable to either the three training phases or the decoding process, which considers only one language and reclusters the word according to the language. If these aspects were modified, the performance could be expected to improve, as it aligns with the idea that a smaller distance between cluster centres leads to a better BLEU score for the NAT model. The experiments of word alignment supervision is ongoing and will be left for future research.

Data augmentation modification The LBPE strategy works well, but it does not combine effectively with the CAMLM adaptor. This could potentially be due to LBPE not being the most effective data augmentation strategy. Attempting to find the best data augmentation strategy while adhering to the principle of dealing with shared character sequences could potentially lead to further improvements.

WDAD work The CAMLM and data augmentation or language embedding layer would deal with either of the issues. However, the combined method would not work synergically. This should be one research direction.

[1] https://iwslt.org/

References

[1] Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5):1–38, 2020. pages 1

[2] Yu Shiwen and Bai Xiaojing. Rule-based machine translation. In *Routledge encyclopedia of translation technology*, pages 186–200. Routledge, 2014. pages 1

[3] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003. pages 1

[4] Nurul Amelina Nasharuddin and Muhamad Taufik Abdullah. Cross-lingual information retrieval. *Electronic Journal of Computer Science and Information Technology*, 2(1), 2010. pages 1

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. pages 1, 2

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. pages 1, 7, 8, 12, 29

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. pages 1, 12

[8] Jiatao Gu and Xiang Kong. Fully non-autoregressive neural machine translation: Tricks of the trade, 2020. pages 2

[9] Yisheng Xiao, Lijun Wu, Junliang Guo, Juntao Li, Min Zhang, Tao Qin, and Tie yan Liu. A survey on non-autoregressive generation for neural machine translation and beyond, 2022. pages 2, 9, 10

[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. pages 2

[11] Yi Ren, Jinglin Liu, Xu Tan, Zhou Zhao, Sheng Zhao, and Tie-Yan Liu. A study of non-autoregressive model for sequence generation, 2020. pages 2

[12] Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement, 2018. pages 3

[13] Chunting Zhou, Graham Neubig, and Jiatao Gu. Understanding knowledge distillation in non-autoregressive machine translation. *arXiv preprint arXiv:1911.02727*, 2019. pages 3, 11

[14] Jiaao Zhan, Qian Chen, Boxing Chen, Wen Wang, Yu Bai, and Yang Gao. Non-autoregressive translation with dependency-aware decoder. *arXiv preprint arXiv:2203.16266*, 2022. pages 3, 12, 17, 18, 27, 28, 30

[15] Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Multilingual training of crosslingual word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 894–904, 2017. pages 3

[16] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining, 2019. pages 3

[17] Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-m: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. *arXiv preprint arXiv:2012.15674*, 2020. pages 3, 14, 20

[18] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2016. pages 3, 23

[19] Tatsuya Komatsu. Non-autoregressive asr with self-conditioned folded encoders. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7427–7431. IEEE, 2022. pages 9

[20] Qinghong Han, Yuxian Meng, Fei Wu, and Jiwei Li. Non-autoregressive neural dialogue generation. *arXiv preprint arXiv:2002.04250*, 2020. pages 9

[21] Arun Babu, Akshat Shrivastava, Armen Aghajanyan, Ahmed Aly, Angela Fan, and Marjan Ghazvininejad. Non-autoregressive semantic parsing for compositional task-oriented dialog. *arXiv preprint arXiv:2104.04923*, 2021. pages 9

[22] Ruchao Fan, Wei Chu, Peng Chang, and Jing Xiao. Cass-nat: Ctc alignment-based single step non-autoregressive transformer for speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5889–5893. IEEE, 2021. pages 9

[23] Minghan Wang, Guo Jiaxin, Yuxia Wang, Yimeng Chen, Su Chang, Hengchao Shang, Min Zhang, Shimin Tao, and Hao Yang. How length prediction influence the performance of non-autoregressive translation? In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 205–213, 2021. pages 9

[24] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. Non-autoregressive neural machine translation, 2018. pages 10, 11

[25] Jason Wei. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6381–6387, 2019. URL `https://arxiv.org/abs/1901.11196`. pages 11

[26] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 649–657, 2015. URL `https://arxiv.org/abs/1509.01626`. pages 11

[27] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.84. URL `https://aclanthology.org/2021.findings-acl.84`. pages 11

[28] Sufeng Duan, Hai Zhao, Dongdong Zhang, and Rui Wang. Syntax-aware data augmentation for neural machine translation, 2020. pages 11

[29] Wei Peng, Chongxuan Huang, Tianhao Li, Yun Chen, and Qun Liu. Dictionary-based data augmentation for cross-domain neural machine translation. *arXiv preprint arXiv:2004.02577*, 2020. pages 11

[30] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Nature*, 1(1):1–16, 2020. URL `https://arxiv.org/abs/2005.14165`. pages 11

[31] Nader Akoury, Kalpesh Krishna, and Mohit Iyyer. Syntactically supervised transformers for faster neural machine translation. *arXiv preprint arXiv:1906.02780*, 2019. pages 12

[32] Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. Non-autoregressive machine translation with latent alignments. *arXiv preprint arXiv:2004.07437*, 2020. pages 12

[33] Yu Bao, Shujian Huang, Tong Xiao, Dongqi Wang, Xinyu Dai, and Jiajun Chen. Non-autoregressive translation by learning target categorical codes, 2021. pages 12

[34] Junliang Guo, Linli Xu, and Enhong Chen. Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 376–385, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.36. URL https://aclanthology.org/2020.acl-main.36. pages 12

[35] Yu Bao, Hao Zhou, Shujian Huang, Dongqi Wang, Lihua Qian, Xinyu Dai, Jiajun Chen, and Lei Li. Glat: Glancing at latent variables for parallel text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8398–8409, 2022. pages 12, 28

[36] IJ Good. Some terminology and notation in information theory. *Proceedings of the IEE-Part C: Monographs*, 103(3):200–204, 1956. pages 12

[37] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006. pages 12

[38] Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. Aligned cross entropy for non-autoregressive machine translation. In *International Conference on Machine Learning*, pages 3515–3523. PMLR, 2020. pages 12

[39] Xiaobo Liang, Lijun Wu, Juntao Li, and Min Zhang. Janus: Joint autoregressive and non-autoregressive training with auxiliary loss for sequence generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8050–8060, 2022. pages 13

[40] Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. *Advances in Neural Information Processing Systems*, 32, 2019. pages 13

[41] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*, 2019. pages 13

[42] Aitor Ormazabal, Mikel Artetxe, Gorka Labaka, Aitor Soroa, and Eneko Agirre. Analyzing the limitations of cross-lingual word embedding mappings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4990–4995, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1492. URL https://aclanthology.org/P19-1492. pages 14

[43] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019. pages 14

[44] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631, 2019. pages 14

[45] Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training, 2021. pages 14

[46] Xiaoze Jiang, Yaobo Liang, Weizhu Chen, and Nan Duan. Xlm-k: Improving cross-lingual language model pre-training with multilingual knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10840–10848, 2022. pages 14

[47] Alessandro Raganato, Raúl Vázquez, Mathias Creutz, and Jörg Tiedemann. An empirical investigation of word alignment supervision for zero-shot multilingual neural machine translation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8449–8456, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.664. URL https://aclanthology.org/2021.emnlp-main.664. pages 14, 15

[48] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003. pages 15

[49] Thorvald Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biologiske Skrifter / Kongelige Danske Videnskabernes Selskab*, 5:1–34, 1948. URL https://www.jstor.org/stable/23329143. pages 15

[50] Suwon Shon, Ahmed Ali, and James Glass. Convolutional neural networks and language embeddings for end-to-end dialect recognition. *arXiv preprint arXiv:1803.04567*, 2018. pages 23

[51] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand, September 13-15 2005. URL https://aclanthology.org/2005.mtsummit-papers.11. pages 27

[52] Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74. Association for Computational Linguistics, 2016. doi: 10.18653/v1/W16-3210. URL http://www.aclweb.org/anthology/W16-3210. pages 27

[53] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling, 2019. pages 27

[54] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`. pages 34