Karlsruhe Institute of Technology

Machine Translation of Arabic Dialects

Bachelor's Thesis of

Ahmad Jayossi

at the Department of Informatics
Institute for AI4LT

Reviewer: Prof. Jan Niehues
Second reviewer: Prof. Alexander Waibel

01. April 2023 – 31. July 2023

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 31.07.2023

. . . . . .

(Ahmad Jayossi)

Abstract

This study investigates how to improve machine translation and dialect identification for Latinised Dialectal Arabic (LDA), using deep neural network models. Latinised Dialectal Arabic, also known as Arabizi, is a way of writing Arabic dialects using Latin script and is often used in online settings. The study looks at two main questions: how well can LDA dialects be identified, and how effectively can LDA be translated into English. Two specific situations were examined. The first is when there is limited parallel data between LDA and English, using transliterated dialectal Arabic as additional data in a supervised Machine Translation approach. The second is when there is no parallel data available, using an supervised and unsupervised Machine Translation approach that relies on pretrained cross-lingual language model utilizing multilingual transfer. Experiments were run on both self-curated and publicly available online datasets. The main findings were that both supervised and unsupervised machine translation models could be improved by using additional data to fine-tune a pretrained cross-lingual model, and a pre-trained BERT model could effectively identify dialects. This research contributes to developing a model for translating LDA into English and understanding how different translation models relate to the Latinised Arabic language.

Zusammenfassung

In dieser Studie wird untersucht, wie die maschinelle Übersetzung und Dialektidentifizierung für lateinisiertes Dialektarabisch (LDA) mit Hilfe von tiefen neuronalen Netzwerkmodellen verbessert werden kann. Latinisiertes Dialektarabisch, auch bekannt als Arabizi, ist eine Art, arabische Dialekte in lateinischer Schrift zu schreiben und wird häufig in Online-Umgebungen verwendet. Die Studie befasst sich mit zwei Hauptfragen: Wie gut können LDA-Dialekte identifiziert werden, und wie effektiv kann LDA ins Englische übersetzt werden. Es wurden zwei spezifische Situationen untersucht. Die erste ist, dass es nur begrenzte parallele Daten zwischen LDA und Englisch gibt, wobei transliteriertes dialektales Arabisch als zusätzliche Daten in einem überwachten maschinellen Übersetzungsansatz verwendet wurde. In der zweiten Situation, in der keine parallelen Daten verfügbar sind, wurde ein überwachter und nicht überwachter Ansatz für die maschinelle Übersetzung verwendet, der sich auf ein vorab trainiertes Cross-Lingual Language Model stützt, das einen multilingualen Transfer nutzt. Die Experimente wurden sowohl mit selbst kuratierten als auch mit öffentlich verfügbaren Online-Datensätzen durchgeführt. Die wichtigsten Ergebnisse sind, dass sowohl überwachte als auch unüberwachte maschinelle Übersetzungsmodelle durch die Verwendung zusätzlicher Daten zur Fine-Tuning eines vorab trainierten Cross-Lingual Language Model verbessert werden können, und dass ein vorab trainiertes BERT-Modell Dialekte effektiv identifizieren kann. Diese Forschung trägt dazu bei, ein Modell für die Übersetzung von LDA ins Englische zu entwickeln und zu verstehen, wie sich verschiedene Übersetzungsmodelle auf die latinisierte arabische Sprache beziehen.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Arabic is a major world language spoken by more than 300 million people across the Middle East, North Africa, and beyond (Horesh and Cotter 2016). Dialectal Arabic (DA) is the informal spoken form of Arabic, which differs significantly in word choice, morphology, pronunciation, and speech tempo, among other aspects, depending on the region where it is spoken. This variation means that there is no standard orthography for DA, making it difficult to write in a way that is easily understandable to speakers from different regions.

In recent years, a new linguistic phenomenon has emerged in the Arab world: Arabizi. This term refers to the transcription of spoken DA in Latin script or Latinised Dialect Arabic (LDA), which allows people to write in their dialects using a standard script that is universally understood. The rise of digital communication in the Arab world has facilitated the growth of Arabizi, which is now used extensively in email and mobile messaging. A 2009 study (Aboelezz 2009) have reported that more than 60% of digital communication in some Arab communities is conducted in Arabizi . This trend is particularly prevalent among young people (Allehaiby 2013).

Translating Arabizi poses a challenge for translators (Zakraoui et al. 2021), given its informal nature and lack of standardization. This is further compounded by the fact that there is a low resource of parallel data available for machine translation systems to use in training (Baert et al. 2020). This makes it difficult to develop accurate translation models for Arabizi, particularly for less commonly spoken dialects. As a result, current machine translation systems are not well-suited for translating Arabizi, and human translators with a deep understanding of the dialect and context are still required.

In light of these challenges, our research investigates the effectiveness of translating Arabizi into English. We aim to develop new techniques for improving the accuracy of machine translation systems for Arabizi, despite the low resource of parallel data and lack of standardization. Our work involves collecting and analyzing a large corpus of Arabizi texts, and developing novel methods for aligning them with their corresponding DA and English translations. We also explore the use of transfer learning techniques, which can leverage knowledge from other languages to improve the accuracy of Arabizi translation.

Overall, our research contributes to a better understanding of the linguistic phenomenon of Arabizi and the challenges it poses for translation. Our goal is that our work leads to the development of more effective translation systems for Arabizi, which facilitates communication and cross-cultural understanding in the Arab world and beyond.

1.2 Research Question

Our primary research question is: *How can machine translation systems be improved for translating LDA into English, despite the lack of standardization and low resource of parallel data available?* This central question is born from the recognition of challenges faced in the translation of less standardized and low-resource languages such as LDA. It also acknowledges the global trend towards digitization and the increasing importance of accurate machine translation to bridge communication gaps.

To address this main research question, we break it down into three sub-questions:

- **RQ1**: *How efficiently can we distinguish LDA dialects?*

  This question is critical as the ability to correctly identify and differentiate between dialects could significantly enhance the translation process's overall accuracy. Furthermore, the dialect-specific nature of LDA increases the complexity of its translation, necessitating a targeted approach. By exploring this question, we hope to lay the groundwork for enhancing dialect-specific translations.

- **RQ2**: *In the case of limited parallel data, how to improve translation quality from LDA into English?*

  In the case of limited parallel data, how to improve translation quality from LDA into English? This question addresses a commonly faced challenge in the field of machine translation - the lack of sufficient parallel data. Limited resources can be a significant obstacle to developing robust translation models. With this research question we try to improve the translation quality under these constrains. With the goal of making Machine Translation for LDA more accessible and effective.

- **RQ3**: *In the case of no existing parallel data, how to improve translation quality from LDA into English?*

  In the case of no existing parallel data, how to improve translation quality from LDA into English? This question pushes the boundary further by imagining a scenario where there is no existing parallel data at all. By exploring innovative methods, such as unsupervised learning or transfer learning from related languages, we aim to expand the possibilities of machine translation even in the most resource-scarce situations.

Each of these sub-questions provides a piece of the puzzle to answer our main research question. By understanding the dialects better (RQ1), finding ways to work with limited data (RQ2), and even no data (RQ3), we hope to uncover techniques to improve the overall quality of LDA to English translations.

1.3 Outline

This thesis is divided as follows: Foundation, Approaches, Evaluation and Ending.

The first section of this thesis provides a theoretical foundation that includes the necessary background, related work, and challenges for the tasks at hand. This foundation will serve as a basis for the subsequent sections of the work.

In the second section, we strive to prepare a stronger foundation for our experiments by gaining a better understanding of the Latinised Arabic language. This will involve the collection of corpora and classification of dialects. Afterwards, we will present the different approaches that we will use for machine translation.

In the third part, we evaluate the experiments that we conducted, which includes the classification of dialects and the various machine translation approaches that we employed. We present our findings and discuss their implications in the context of the research question that we have posed.

Finally, we summarize our work and present our ideas for future research. Including a discussion of the limitations of our study and suggestions for areas of inquiry that could be pursued in order to build upon our findings.

2 Background

This chapter establishes the foundational knowledge requisite for our research, consisting of three important sections.

The first section is dedicated to providing a detailed linguistic background. Herein, we delve into the diverse dialects of Arabic, their variations, and their Latinised forms. Furthermore, we will outline the challenges inherent in identifying Latinised Dialectal Arabic (LDA) and translating it. To conclude this section, we will survey the data sets available that could prove beneficial for our experimental undertakings and subsequent evaluations.

The second section provides a comprehensive overview of the classification problem, with a specific focus on dialect identification. We also introduce the BERT (Bidirectional Encoder Representations from Transformers) technology, a crucial tool in our research, which will be utilized to train our dialect classification model.

The final section goes into a detailed exploration of the Neural Machine Translation approach, describing its supervised and unsupervised methods. This section intends to augment our understanding of how machine learning can aid in translation tasks, thereby enabling a more nuanced analysis of our research.

2.1 Linguistic Background

In this section, we introduce the multiple varieties of regional Arabic dialects and their Latinised version.

2.1.1 Arabic Dialects

Arabic is a complex and fascinating language with a rich linguistic history. Spoken by over 300 million people across more than 20 countries, it is a vital part of the cultural and linguistic landscape of the Middle East and beyond (Horesh and Cotter 2016). The Arabic language can be broadly divided into two parts: Modern Standard Arabic (MSA) and a variety of Dialectal Arabic (DA).

MSA is a standardized form of Arabic with a set of grammar rules that is used in formal settings, including written communication, television broadcasts, and official speeches. It is taught in schools and used in higher education. In contrast, DA is an informal and mostly spoken form of Arabic. Although there have been an attempt to standardize the spelling of some Arabic dialects (Habash et al. 2012), it remains lacking standardization and grammar rules.

While MSA is used in formal contexts, DA is used in day-to-day spoken communication. With the rise of online communication platforms such as chat rooms, blogs, and social

media, we have seen an increase in the use of DA in written form as well. This has contributed to the creation of a new type of Arabic, sometimes referred to as "Arabizi," which blends DA with elements of the Latin alphabet and English words.

One of the most unique aspects of Arabic is the wide range of dialects spoken across different regions. Each dialect has its own distinct vocabulary, pronunciation, and grammar. For example, the word for "car" in Egyptian Arabic is "araba" while in Levantine Arabic it is "sayyara" and "Karhaba" in Tunisian Arabic. These variations are a result of the historical and regional influences on the development of the language. The vastness of the Arab world means that there are many different dialects, each with its own unique features.

Arabic Dialect Varieties

According to (Zaidan and Callison-Burch 2014), Arabic dialects can be categorized into several regional groups:

- **Maghrebi**: This dialect is spoken in North African countries, including Morocco, Algeria, Tunisia, and Libya. It is heavily influenced by the Berber and French languages, resulting in a distinct pronunciation and vocabulary. Maghrebi Arabic is generally only understood within this region (Berrimia et al. 2020).
- **Egyptian**: This dialect is widely recognized as the most popular Arabic dialect due to the significant influence of Egyptian media on the Arab world, and the fact that Egypt is the most populous Arab country (Haeri 2003).
- **Gulf**: This dialect encompasses the dialects of the Gulf countries, including Saudi Arabia, Kuwait, Qatar, Bahrain, United Arab Emirates, and Oman. The Gulf dialect has a close relationship with Modern Standard Arabic, as its the birthplace of the Arabic language (Al-Jallad 2014).
- **Levantine**: This dialect set covers the dialects of the Levant countries, namely Palestine, Jordan, Lebanon, and Syria. Levantine Arabic is known for its distinctive pronunciation.
- **Iraqi**: This dialect is specific to Iraq and some neighboring regions.

2.1.2 Latinised Arabic

With the advent of globalization and technology, the Arab world gained access to the internet. However, the initial versions of Internet Explorer, Windows Mobile, and Android did not support Arabic display (Darwish 2014). As a result, many Arabs resorted to communicating online using Latin script, leading to the emergence of Latinised Dialect Arabic. This form of communication gained popularity among Arab youth in several countries (Allehaiby 2013; Bianchi 2012).

LDA is sometimes referred to as Arabizi, a blend of the words "Arabi" (Arabic) and "Englizi" (English) (Farrag 2012; Yaghan 2008) we will be using the words LDA and Arabizi interchangeably in this work. According to research done by (Aboelezz 2009; Alabdulqader

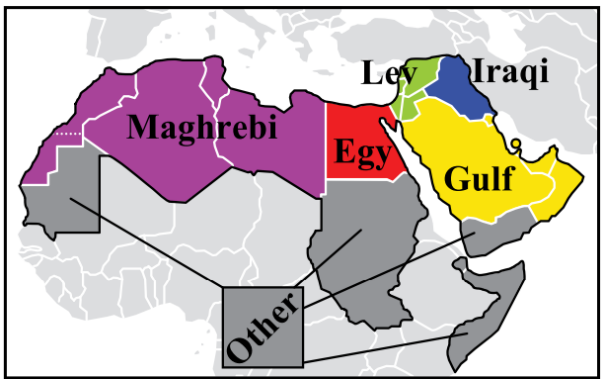

Figure 2.1: Dialect groups according to Zaidan and Callison-Burch 2014

et al. 2014; Allehaiby 2013; Gibson 2015; Jaran and Al-Haq 2015; Keong et al. 2015; Tobaili 2016; Yaghan 2008), the usage of LDA varies among Arabs of different ages, genders, and educational levels. 2.1.2 summarizes the percentages of LDA usage among various Arab demographics.

Reference	Year	Location	Participants	Data	Size of Data	Arabizi	English	Arabic
(Keong et al. 2015)	2015	Malaysia	20 Arab Post Graduates	SMS	200 Messages	35%	50%	10%
(Bies et al. 2014)	2014	Egypt	26 Native Arabic Speakers	SMS	101,292 Messages	77%	-	23%
(Alabdulqader et al. 2014)	2014	Saudi Arabia	61 Students and Non-students	SMS	3236 Messages	15%	8%	74%
(Bianchi, 2012)	2012	Jordan	-	Online Forum	460,220 Posts	35.5%	17.5%	32%
(Al-Khatib and Sabbah 2008)	2008	Jordan	46 Students	SMS	181 Messages	37%	54%	9%

Table 2.1: Percentage of Arabizi usage according to previous work Tobaili 2016

LDA is a form of transliteration where dialectal Arabic is represented using Latin script and numbers instead of Arabic letters. The use of numbers compensates for the missing Arabic letters as there are more letters in Arabic than Latin, and they need to be accounted for (As it can be observed in Table 2.2). However, it's important to note that the numbers that substitute for missing letters are region-dependent and represent different letters in various countries. For instance, in Tunisia, 9 represents the Arabic letter ق (qāf), while in Palestine, it represents the letter ص (sād).

2.1.3 LDA challenges for Dialect Identification

1. **Coexistence of Modern Standard Arabic and Dialectal Arabic:** Arabizi is characterized by the simultaneous presence of Modern Standard Arabic and Dialectal Arabic. This convergence poses difficulties in distinguishing between the two, as code-switching between these forms is a common feature. Notably, speakers often

Arabic Letter	Name	Arabizi	Phoneme	Example in Arabizi w/ Translation
ح	ha’	7	ħ	sa7eb (Friend)
خ	kha’	5	x	5alas (Enough)
ع	’ayn	3	ʕ	3arabi (Arabic)
غ	ghayn	8	ɣ	8arib (Strange)
ء	hamzah	2	ʔ	2ana (I/Me)
ص	sād	9	$s^ʕ$	ma9la7a (Interest)
ض	dād	9’	$d^ʕ$	9’aye3 (Lost)

Table 2.2: Numerical representation of substitution numbers used in Arabizi Levantine/Egyptian version

employ MSA vocabulary alongside dialectal expressions, further complicating the process of dialect identification.

2. **Code-switching and Emojis** The phenomenon of code-switching, as observed in dialectal Arabic, adds an additional layer of complexity to dialect identification (Samih et al. 2016). Code-switching refers to the alternation between different languages or language varieties within a single conversation or text. For example the English word ’men’ could be interpreted in LDA as "mn" من meaning ’from’ or "myn" مين meaning ’who’ (Shazal et al. 2020).

3. **Limited Datasets and Lack of Representation:** A significant impediment in dialect identification within Arabizi is the scarcity of labeled datasets that predominantly focus on Egyptian and Levantine dialects. However, this emphasis on specific dialects results in a lack of representation for other dialectal variations. As a consequence, the underrepresentation of non-Egyptian and non-Levantine dialects compromises the overall accuracy and applicability of dialect identification models. The limited availability of diverse and comprehensive labeled datasets hampers the development of accurate and generalizable models, thereby limiting their effective utilization in broader linguistic contexts

2.1.4 LDA challenges for Machine Translation

1. **Differentiating English and Arabizi: Orthographic Variability:** similar to the second challenge in Dialect Identification 2. LDA, being a hybrid form of Arabic writ-

ten in the Latin script, lacks a standardized orthographic structure. This variability in spelling conventions makes it challenging to rely solely on an English dictionary to identify English words in Arabizi. Some Arabizi words and English words may share identical spellings, further exacerbating the difficulty of distinguishing between the two languages Darwish 2014.

2. **Lack of Spelling Conventions and Building an Arabizi Dictionary:** Arabizi, along with Arabic dialectal text, lacks established spelling conventions. Creative spelling variations are prevalent, making it prohibitive to construct a comprehensive dictionary of Arabizi words. The absence of standardized spellings in Arabizi poses a significant obstacle to the development of linguistic resources and tools for accurate Arabizi analysis and processing. For example the Arabic word حبيبي Hbyby 'My beloved' has many different spellings: habibii, hbebe, habibi, 7abiby, 7abibi, hbebee, 7abeby, 7biby, 7bibi and many more... Shazal et al. 2020

3. **Variations in Arabizi Dialectal Differences** Different Arabic dialects may employ alternative words altogether, further complicating the translation and interpretation of specific terms. For example, the Tunisian dialect uses 'bhim' to refer to a donkey, while the Egyptian dialect uses 'ga7sh' and Palestinians use '7mar'. These dialectal variations in Arabizi pose challenges for accurate understanding and analysis, requiring comprehensive linguistic resources and context-aware approaches.

4. **Limited Availability of LDA Language Resources** Publicly available Arabizi parallel corpora, which provide aligned texts in both Arabizi and English, are rare. Most existing resources primarily focus on Egyptian and Levantine dialects, limiting the availability of diverse data for training machine translation models and other language processing tasks. The scarcity of comprehensive language resources hinders the development of accurate computational models and restricts their applicability to dialects beyond the Egyptian and Levantine regions.

2.1.5 Available Datasets

Several datasets have been developed for LDA, but due to the scarcity of parallel corpora, the number of such resources remains limited. In the case of Egyptian LDA, several datasets have been collected, including Bies et al. 2014; Chen et al. 2017; Tobaili 2016; Tracey et al. 2021. Similar datasets for other Arabic dialects, such as Lebanese Tobaili 2016, Algerian Guellil et al. 2017, and Tunisian Masmoudi et al. 2019; Younes et al. 2015 are also available.

Of these, the Tracey et al. 2021 dataset is the most relevant for our purposes, as it is the only one that provides parallel corpora for Egyptian LDA and English. This dataset

comprises SMS/Chat Parallel Training Data among speakers of Egyptian LDA, along with their corresponding translations into English. It consists of approximately 723,000 tokens of Egyptian Arabic, making it a valuable resource for training and evaluating machine translation systems.

In addition to the above-mentioned datasets, several other resources have been utilized for improving machine translation systems for LDA. These include:

LDC2012T09 which we will call (DA_{eg_lev}) was developed by (Raytheon 2012), is a corpus consists of Levantine and Egyptian dialectal Arabic web text obtained from various sources. Firstly, a large amount of Arabic text was automatically filtered from web sources, such as weblogs and online user groups, totaling around 350 million words. Non-Arabic and Modern Standard Arabic (MSA) words were removed, resulting in a subset of approximately four million words. Additionally, Arabic dialect web sites were manually harvested by Sakhr Software. To classify the passages, annotators from categorized them as MSA or regional dialects. Only Levantine and Egyptian passages were further processed, including sentence segmentation and translation into English. The resulting parallel corpus includes Egyptian and Levantine dialects containing around 38k sentences in the Egyptian dialect and circa 138k sentences of Levantine dialect.

The Modern Standard Arabic dataset from (Lison and Tiedemann 2016)[Opensub] will be used as auxiliary data, and the MultiDial dataset for testing purposes. However, it should be noted that these datasets are in Arabic script and primarily comprise Egyptian and Levantine dialects. As they do not provide parallel text for LDA and English, they need to be transliterated before being used in machine translation tasks.

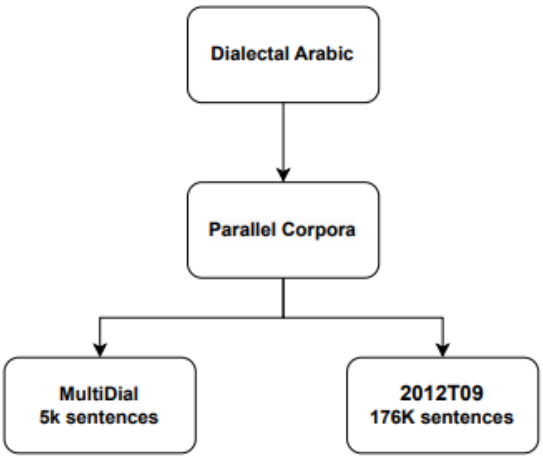

Figure 2.2: Available Dialectal Arabic datasets

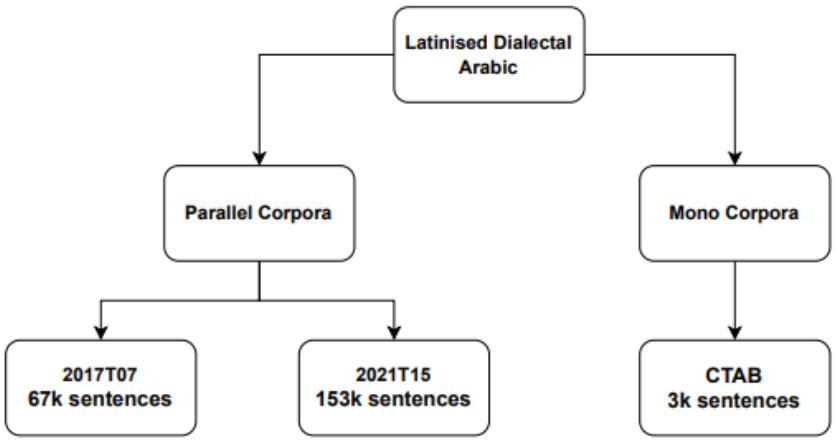

Figure 2.3: Available LDA datasets

2.2 Classification

Classification tasks are fundamental in natural language processing (NLP). In a classification task, the goal is to assign an input instance to one or more predefined categories or classes based on its features. This brief introduction will provide an overview of the basics of classification tasks, focusing on dialect classification / Identification in NLP.

2.2.1 Dialect Identification

Automatic dialect identification using neural networks involves building a machine learning model that can identify the dialect or variant of a language spoken by a speaker from a given audio or text input using neural networks.

Although Dialect or language identification is a core task in natural language processing it remains a difficult task (Caswell et al. 2020). Specifically the subtask of distinguishing between similar languages or dialects (Zampieri et al. 2014) More specifically, the model takes as input a speech or text sample in a language, and processes it through a series of layers of neural networks, typically using a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) or transformer networks, to learn the patterns and features that distinguish one dialect from another. The model is trained on a large dataset of labeled speech or text samples from different dialects, where each sample is labeled with its corresponding dialect or variant.

The output of the model is a probability distribution over the set of dialects or variants in the training data, where the highest probability indicates the most likely dialect or variant spoken in the input sample. The model can be fine-tuned or adapted to new dialects or languages by training it on additional labeled data from these dialects or languages.

2.2.2 BERT-based Classification

BERT, or Bidirectional Encoder Representations from Transformers, is a groundbreaking NLP model developed by Google. It has significantly improved the performance of various NLP tasks, such as sentiment analysis, question-answering, and text classification. BERT's key innovation lies in its architecture. The architecture of BERT is centered around the transformer model, initially proposed in the paper by (Vaswani et al. 2017). The transformer model uses self-attention mechanisms to understand the context of words in texts, allowing it to capture contextual relations between words.

In the case of BERT, it specifically utilizes a stack of these transformers, creating a deep bidirectional model. This means BERT is looking at the context from both sides (left and right of a word) simultaneously, unlike traditional left-to-right or right-to-left unidirectional models. The model consists of two main parts: the encoder and the classifier.

The encoder (seen in Figure 2.4) part is responsible for understanding the input text. It is composed of several identical layers, each containing two sub-layers: a multi-head self-attention mechanism, and a position-wise fully connected feed-forward network. A residual connection is employed around each of the two sub-layers, followed by layer normalization. The original BERT paper (Devlin et al. 2018) describes two bert architecture sizes $BERT_{Large}$ and $BERT_{Base}$. The latter consists of 12 Transformer layers, with each layer having a hidden size of 768. It uses 12 attention heads for the multi-head self-attention mechanism within each Transformer layer. This configuration results in a model with approximately 110 million total parameters.

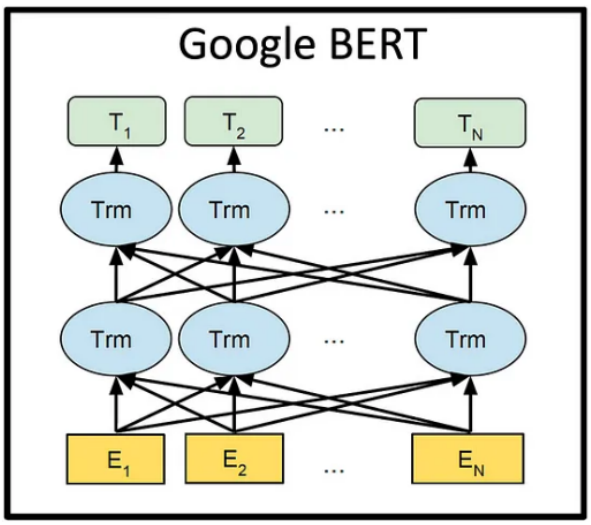

Figure 2.4: Illustration of BERT's Bidirectional Transformer Devlin et al. 2018

The classifier part is usually a simple linear layer that sits on top of the encoder output for the [CLS] token (special classification token that's added to the input). It generates the final predictions and scores for classification tasks. However, the success of BERT models is not just due to its architecture, but also due to the pretraining and fine-tuning processes.

BERT pretraining forms a crucial part in the construction of BERT models. It involves training the model on a substantial corpus of text data, enabling it to learn the underlying linguistic structures and patterns. This pretraining phase relies on unsupervised learning and consists of two primary tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), both of which were introduced in the original BERT paper (Devlin et al. 2018).

In the MLM task, certain words or tokens within a sentence are randomly masked or hidden from the model during training. The model is then tasked with predicting these masked words based on the context provided by the non-masked words in the sentence. This enables the model to learn a sense of language structure and context, improving its ability to understand and generate appropriate responses within contextually similar scenarios.

NSP, on the other hand, focuses on understanding the relationship between sentences. In this task, the model is presented with pairs of sentences and must predict whether the second sentence logically follows the first in the original document. This helps the model to comprehend the broader, discourse-level relationships between sentences, a crucial skill for tasks like document summarization and question answering.

Fine-tuning is the process of adapting the pretrained BERT model to a specific NLP task, such as text classification. During fine-tuning, the model is trained on a smaller, task-specific dataset, allowing it to learn the nuances and patterns relevant to the task at hand. In this process BERT adjusts its learned representations to a specific task with labeled data, ultimately resulting in a model that is highly specialized for the target task (Sun et al. 2019). The combination of pretraining and fine-tuning has made BERT-based classification models highly effective and widely used in various NLP applications.

2.3 Neural Machine Translation

Neural machine translation is a technology used for automatic text translation from one language to another using artificial neural networks. It is considered the successor of Statistical machine translation (SMT) and has several advantages over its predecessor. NMT utilizes deep learning, which is a subset of machine learning that uses multi-layer neural networks to learn and improve on a task. Unlike SMT, NMT utilizes vector representation for words and internal states, which enables the model to learn semantic relationships between words in a sentence (Bahdanau et al. 2014). Additionally, NMT utilizes representational learning, which is the automatic extraction of useful features from raw data. This allows the model to extract features from the input text that are relevant to the translation task, resulting in more accurate translations. Overall, NMT has proven to be a significant improvement over SMT in terms of translation quality (Junczys-Dowmunt et al. 2016), and it has become the primary approach for machine translation tasks.

In recent years, a new approach to NMT has emerged, which is based on the Transformer architecture. The Transformer is a neural network model designed specifically for sequence-to-sequence tasks such as machine translation. It was introduced in the paper "Attention is All You Need" by (Vaswani et al. 2017) and has since become the foundation for many state-of-the-art NMT systems.

2.3.1 Supervised Machine Translation

A bilingual supervised translation model trains using parallel data from language X to language Y. For N source sentences $X = x_1, x_2, \ldots, x_N$ there are target language sentences $Y = y_1, y_2, \ldots, y_N$. From a probabilistic standpoint, it can be framed as the search for a target sentence, denoted as 'y,' which maximizes the conditional probability of 'y' given a source sentence, denoted as 'x.' Mathematically, this can be represented as $\arg\max_y p(y \mid x)$.

Our objective is to train a parameterized model that maximizes the conditional probability of sentence pairs using a parallel training corpus. By leveraging this corpus, we aim to learn the underlying conditional distribution in the translation model. Once this distribution has been acquired, we can generate a corresponding translation for a given source sentence by searching for the sentence that maximizes the conditional probability (Bahdanau et al. 2014).

In the vast domain of supervised machine translation models, our thesis chooses a Transformer-based variant, given its distinguished efficacy.

Transformer Architecture

As depicted in Figure 2.5, the Transformer architecture divided into two core components: the encoder on the left, and the decoder on the right. The following subsection explains these components in depth.

Encoder Block

- Input Preparation

  The initial task in the encoding process is to convert the input words into numerical values, thus facilitating their comprehension by the machine. This is achieved through the process of embedding, wherein words are transformed into vectors in an embedding space. In this space, vectors representing similar words are positioned close to each other. Subsequently, positional encoding is performed, generating for each word vector a context of its location within the text or sentence, thereby capturing the nuances of word meanings depending on their positional context. The resultant context, or the transformed input, is then ready to be given into the encoder block.

- Multi-Head Attention

  The initial phase of the encoder block employs an attention mechanism, generating an attention vector for each word in the sequence. These vectors depict the relevance of a word in relation to other words within the sequence. The essence of the multi-head attention mechanism lies in its ability to allow each head to identify and learn different patterns within the sequence, thereby increasing the expressiveness and the ability of the Transformer model to learn complex relationships. Once the various attention vectors for each word have been generated, a weighted average of these vectors is computed.

- Feed-Forward Neural Network (FFN)

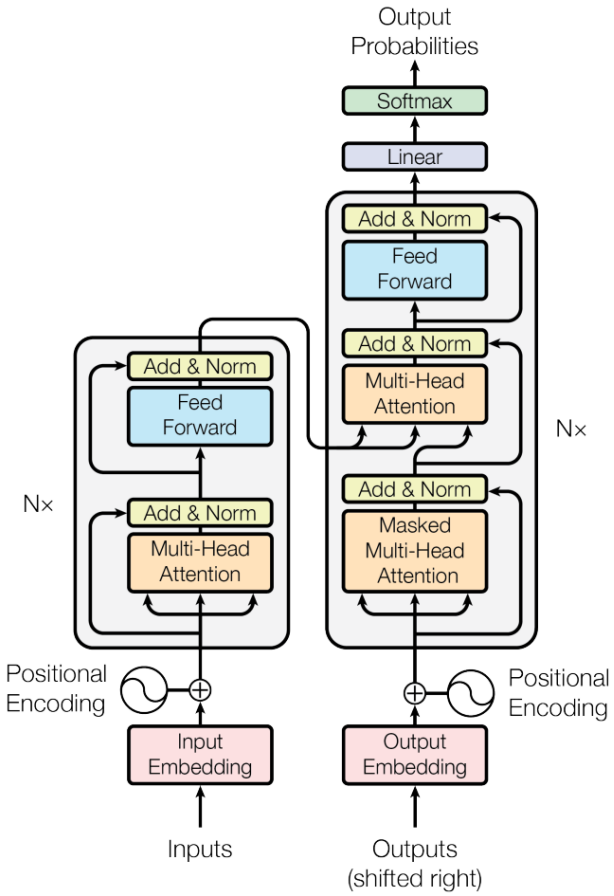

Figure 2.5: Illustration of the Transformer model architecture Vaswani et al. 2017

The FFN is employed to transform the attention vectors, which are the outputs from the preceding step, into a format that can be readily accepted by the following encoder/decoder layer.

Decoder Block

Assume a scenario where an English-Arabic translation model is being trained using parallel English and Arabic sentences. Herein, the English sentences are processed by the encoder block, while the Arabic counterparts are fed to the decoder block.

- Input Preparation

  similarly to the operations in the encoder block, words in the input sentences are transformed into context vectors through a process involving embedding and positional encoding.

- Masked Multi-Head Attention

  This step is comparable to the corresponding phase in the encoder block, with the key distinction being that this version employs masking. The masking process prevents the attention heads from prematurely viewing future words in the sequence.

Therefore, the decoder doesn't receives a full sentence, but a masked sentence, and its objective is to generate a translation of the sentence initially input to the encoder.

- Multi-Head Attention

  Upon generation of the attention vectors from the masked multi-head attention layer and the encoder, these vectors are input to the Multi-head attention block. This block is responsible for the mapping of English and Arabic words, learning their connection, and subsequently outputting attention vectors for every word in the respective English and Arabic sentences.

- Feed-Forward Neural Network

  Upon the generation of attention vectors in the previous step, these vectors are processed by a Feed-Forward Neural Network. This transformation helps in the better capture of local information or relationships between words and their immediate neighbors.

- Linear Layer & Softmax

  Finally, the output from the FFN is processed through a linear layer, transforming the high-dimensional output from the previous layer into a vector of a size equal to the number of words in the output vocabulary. Each element in this vector corresponds to a specific word. A softmax function is then applied to convert this vector into a probability distribution. The word associated with the highest probability is chosen as the translation of the input word.

2.3.2 Unsupervised Machine Translation

According to the paper (Lample, Conneau, et al. 2018) the unsupervised machine translation model using monolingual corpora only can be defined as follows:

Unsupervised Machine Translation (UNMT) is the training of a translation model without parallel corpora of the languages to be translated from and to. Instead, sentences from monolingual corpora of both languages are used to map them to a common latent space.

The model is first pre-trained to create an initial embedding of both languages (Conneau et al. 2017). The translation model comprises two important elements, the encoder and decoder. The encoder, explained in depth in 2.3.1, takes an input sentence $x = x_1, ..., x_n$ and generates hidden states $h = h_1, ..., h_n$ using the pre-trained word embeddings. These hidden states are vectors in a common latent space. The decoder, as explained in the previous section 2.3.1, on the other hand, takes the output of the encoder and outputs a sequence of words in the target language.

The training of the translation model comprises multiple training objectives:

- **Denoising Auto-Encoding**: The model learns to reconstruct sentences in a language from a noisy version of it. It takes a noisy input sentence, where noise refers to slightly shuffled words and random word dropping, encodes it, and tries to reconstruct/decode it in the same language, see Figure 2.6. This will teach the encoder

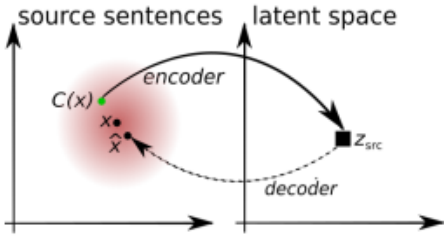

Figure 2.6: Denioising Auto-Encoding (Lample, Conneau, et al. 2018)

and decoder to map the source and target languages since the latent space is shared. The objective function is defined as:

$$L_{auto}(\theta_{enc}, \theta_{dec}, Z, L_1) = \mathbb{E}(\Delta(x, \hat{x}))$$
$$L_{auto}(\theta_{enc}, \theta_{dec}, Z, L_2) = \mathbb{E}(\Delta(x, \hat{x}))$$
$$\hat{x} = \text{decode}(\text{encode}(\text{noise}(x), L))$$

Where $\theta_{enc}, \theta_{dec}$ are the encoder and decoder parameters. Z is the set of words embedding of the source and target languages. $\hat{x}$ is the reconstruction of a noisy version of x. δ is the measure of the difference between the two sequences.

- **Cross Domain Training:** The model is iteratively trained to translate from one language to another, see Figure 2.7. The translation model is updated after each iteration. $y = TM_i(x)$. The main objective of this function is to be able to translate sentences from the dataset of l_1 to l_2 and vice versa. Initially sentence x from l_1 is given with the goal to generate a corrupted translation of it in l_2. This allows the encoder and decoder to learn to reconstruct x from corrupt y where $y = TM(x)$

$$L_{CD}(\theta_{enc}, \theta_{dec}, Z, L_1, L_2) = \mathbb{E}(\Delta(x, \hat{x}))$$
$$\hat{x} = \text{decode}(\text{encode}(\text{noise}(x), y), L_1)$$
$$L_{CD}(\theta_{enc}, \theta_{dec}, Z, L_2, L_1) = \mathbb{E}(\Delta(x, \hat{x}))$$

After each step, TM_i is updated.

- **Adversarial Training**: In this objective, we train the discriminator. The discriminator classifies between the encoding of source and target sentences (Ganin et al.

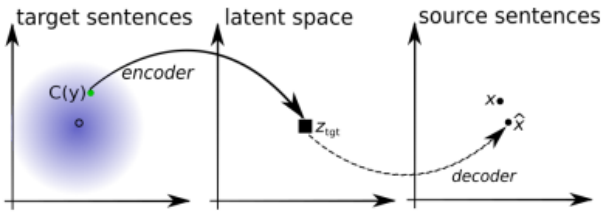

Figure 2.7: Denioising Cross Domain (Lample, Conneau, et al. 2018)

2016), while the encoder is training to fool the discriminator.

$$L_{adv}(\theta_{enc}, \mathcal{Z}|\theta_{disc}) = -\log(\text{prob}(L_1|\text{encoder}(x, L_2)))$$

- **The final objective function:**

$$\begin{aligned}\mathcal{L}(\theta_{enc}, \theta_{dec}, \mathcal{Z}) =& \lambda_{auto}[\mathcal{L}_{auto}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, L_1) + \mathcal{L}_{auto}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, L_2)] \\ &+ \lambda_{cd}[\mathcal{L}_{cd}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, L_1, L_2) + \mathcal{L}_{cd}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, L_2, L_1)] \\ &+ \lambda_{adv}[\mathcal{L}_{adv}(\theta_{enc}, \mathcal{Z}|\theta_{discriminator})]\end{aligned} \tag{2.1}$$

where λ_{auto}, λ_{cd}, and λ_{adv} are hyperparameters that control the relative importance of each loss term.

In conclusion, unsupervised machine translation is a technique that allows us to train translation models without the need for parallel corpora. Instead, it leverages monolingual data to learn a shared latent space, which enables the model to translate between languages. The training process involves multiple objectives, including denoising auto-encoding, cross-domain training, adversarial training, and a final combination of all objectives. The resulting translation model can be used to translate between languages even when parallel corpora are not available, making it a useful tool for our case (Koneru et al. 2021).

2.4 Transliteration

Transliteration involves converting the letters of one alphabet (or script) into the equivalent characters of another. Unlike translation, which aims to convey the meaning of text from one language to another, transliteration focuses on representing the phonetic characteristics of the original text. This ensures the text remains pronounced approximately the same across languages.

$$\text{Transliteration} : \text{Source Alphabet} \rightarrow \text{Target Alphabet} \tag{2.2}$$

In simpler terms, transliteration helps speakers of one language read and pronounce words from another language without needing to understand the source language itself. For instance, an Arabic word transliterated into English would still sound like the original when read by an English speaker, regardless of its meaning.

2.4.1 Transliteration in Arabic Language

Arabic, as a Semitic language, possesses a unique alphabet that can be challenging for speakers of other languages to read or pronounce accurately. Hence, Arabic transliteration becomes crucial for non-Arabic speakers to accurately articulate Arabic terms.

Arabic transliteration mainly converts Arabic characters into Latin script, the most widely used script globally. It takes into account the specific phonetic qualities of Arabic, that do not have direct equivalents in many other languages.

Given the significant differences in sounds and phonetic structures between Arabic and other languages like English, several transliteration schemes have been developed. The most common ones include Buckwalter Transliteration and the ISO 233 system.

The choice of a transliteration system often depends on the specific needs of the work, such as whether accuracy, ease of use, or recognition for Arabic speakers is most important. Some systems prefer to preserve the phonetic accuracy of Arabic, leading to complex systems with many diacritical marks, while others aim for simplicity, even if it results in some loss of phonetic accuracy.

2.5 Evaluation Metric BLEU

In the assessment of our experimentation translation model's quality, the BLEU (Bilingual Evaluation Understudy) (Papineni et al. 2002) metric is employed as an evaluation mechanism. This automatic metric facilitates the rating of MT systems, assessing their performance through a comparative analysis between the system-generated output sentences and their respective gold standard reference translations.

BLEU aims to measure the quality of machine translations by determining the similarity between the machine-generated translations and human translations, referred to as "reference sentences". To provide a comprehensive understanding of a model's performance, an average is calculated from the individual evaluations of all the generated translations. This method effectively calculates an approximate measure of the translation model's overarching quality.

BLEU utilizes the concept of n-gram precision to quantify the degree of similarity between a generated sentence and its reference translation. The n-gram precision is determined by identifying sequences of n consecutive words that are congruous in both the machine and human translations.

It is crucial to underscore an integral component of BLEU evaluation, known as the brevity penalty (BP). This functionality impose a score deduction for machine-generated sentences that are excessively short. The purpose of the brevity penalty is to discourage MT systems from producing inordinately short translations, which often results in loss of meaning. The extent of the penalty escalates in proportion to the length disparity between the machine-generated and reference sentences, thereby ensuring a balance between conciseness and information completeness.

The BP formula can be expressed as

$$BP = \begin{cases} 1 & \text{if } g > r \\ e^{1-\frac{r}{g}} & \text{if } g \leq r \end{cases}$$

where g denotes the length of the generated translation and r is the length of the reference sentence.

The complete BLEU formula:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \cdot \log(p_n)\right)$$

In this formula, w_n are the weights for each n-gram (usually equal when calculating the geometric mean), and p_n are the precisions for each n-gram.

3 Related Work

This chapter delves into the subject of Dialectal Arabic and its latinised variant, explaining its relation to our main thesis. In the first section we present the related scientific work in dialectal Arabic including dialect identification and the translation of dialectal Arabic then the related work that has been done to LDA.

In the first part of the LDA section, we examine the research that has been conducted on the specific task of language and dialect detection, offering insights into this complex field. Moving forward, the second part delves into the transliteration of Latinised dialectal Arabic, an area where significant research has already been conducted. The third and final part investigates various strategies used in translating Latinised dialectal Arabic. This section also looks into efforts made to improve the accuracy of this translation process.

3.1 Dialectal Arabic in NLP

3.1.1 Identification of Arabic Dialects

The following studies demonstrate the importance of identifying and classifying Arabic dialects and LDA in NLP tasks. For our work on machine translation and classification of Latinised dialectal Arabic, understanding and addressing the challenges related to dialect identification and Arabizi detection will be crucial for developing an effective classification and translation system.

In the field of Natural Language Processing, several studies have focused on the identification and classification of Arabic dialects. (Zaidan and Callison-Burch 2014) achieved an impressive 85.7% accuracy in detecting the dialect of Arabic sentences using n-gram probabilistic classifiers. (Salloum et al. 2014) introduced a sentence-level classification approach that improved machine translation system selection by 1.0% BLEU score. (Malmasi et al. 2015) used a combination of linear Support Vector Machine and a meta-classifier to achieve 74% accuracy in a 6-dialect classification. (Lulu and Elnagar 2018) employed deep neural network models and found LSTM to be the superior model with an accuracy rate of 71.4%.

In recent research, (Talafha et al. 2020) investigated the use of BERT for country-level dialect identification, achieving 42.86% accuracy on 21 dialects by leveraging pretraining on a large dataset of tweets. These studies highlight the importance of dialect identification in various NLP tasks, including machine translation system selection and understanding the regional variations of Arabic.

Additionally, researchers have also focused on Arabizi detection, which involves identifying Arabizi in texts. (Darwish 2014) and (Tobaili 2016) developed approaches for word-level and sentence-level Arabizi detection, respectively, using language models,

statistical sequence labeling algorithms, and sentence-level features. (Saâdane et al. 2018) addressed the identification of North African dialects, including Algerian, Tunisian, Moroccan, and Egyptian dialects, using a dictionary-based approach. (Adouane et al. 2016) aimed to differentiate Arabicized Berber from other Arabic varieties, while (Shazal et al. 2020) presented a character-level Seq2Seq model for Arabizi detection.

3.1.2 Machine Translation for Dialectal Arabic

The study (Jeblee et al. 2014) illustrates a strategy involving a consequential process of translation from English to MSA, followed by a translation to Egyptian Arabic. This approach is based on the existence of reliable English to MSA translation models, and has successfully yielded a BLEU score of 42.9, which demonstrates its potential efficacy. This approach provides an established method of translating languages into dialectal Arabic, albeit not specifically Latinised dialectal Arabic. However, the concept of sequential translation may offer valuable insights for developing our own machine translation model.

In the work (Farhan et al. 2020) the authors proposed two systems to mitigate the Arabic dialect translation issue with dealing with large amounts of vocabulary. The first system is Dialectal to Standard Language Translation (D2SLT) and the second system is based on google NMT (GNMT). Their methods and findings can contribute to our understanding of strategies used in translating dialectal Arabic and improving the accuracy of the translation process. Their work is especially significant, as their strategies in dealing with the unsupervised NMT of dialectal Arabic and large vocabularies could prove beneficial to improve the accuracy when translating Latinised dialectal Arabic.

In the context of datasets and experimental research, (Zbib et al. 2012) work offers significant insights. They curated a dataset comprising Levantine and Egyptian dialects, and conducted multiple machine translation experiments. Their work shows the importance of morphological analysis and cross-dialect training, and they demonstrate the effectiveness of translating from dialectal Arabic to MSA, then to English. Their research also shows that better results can be obtained from a system trained with both MSA and Dialectal Arabic. The insights from this research align closely with our work, as it demonstrates the importance of incorporating both MSA and dialectal Arabic in the training process. Moreover, their emphasis on morphological analysis could be relevant when dealing with the complexities of Latinised dialectal Arabic.

3.2 Latinised Dialect Arabic in NLP

3.2.1 Transliteration

(Sajjad et al. 2012) presented a novel approach to Arabizi transliteration mining by combining transliteration and non-transliteration models. Expectation Maximization is employed to learn parameters and determine word pairs as transliterations based on probabilities assigned by the transliteration sub-model. Further enhancement is achieved by incorporating probability estimates from unlabeled data, resulting in parameters that closely align with those estimated from labeled data. The approach provides an effective strategy for

transliteration mining, which can inform our development of a machine translation system for Latinised dialectal Arabic. The approach is relevant, as it provides an effective strategy for transliteration mining, which can inform our development of a machine translation system for Latinised dialectal Arabic.

In a separate study, (Chalabi and Gerges 2012) developed a hybrid approach to constructing a Romanized Arabic transliteration engine, which was subsequently extended to cover other scripts. This approach builds upon previous work and relies on statistical machine translation techniques. Their work contributes a methodological perspective that could be applied in our research context.

(Al-Badrashiny et al. 2014) introduced a system that utilizes a character-level finite state transducer to generate transliterations for Arabizi words. The generated transliterations are filtered and selected using a language model, leading to an improved accuracy of 69.4% compared to a previously proposed method, which achieved 63%. The concept of character-level transliteration could be advantageous in our research for dealing with Latinised dialectal Arabic.

(Eskander et al. 2014) presented novel modules for the detection of foreign words (Non-Arabic words written in Latin) as well as of emoticons, sounds, punctuation marks, and names in Arabizi. then transliterates Arabizi into Arabic script. Their methods provide a foundation for dealing with or filtering out with non-standard and non-textual elements in Latinised dialectal Arabic.

Another study by (Masmoudi et al. 2019) focuses on the conversion of Latin Tunisian Arabic into Arabic script. The researchers propose two models: a rule-based model that employs a set of conversion rules for transforming Tunisian dialect Arabizi text into Arabic script, and a discriminative model that addresses sequence classification tasks. The rule-based model gives us important insights for our research, as it might allow us to handle specific dialects in Latinised dialectal Arabic.

(Guellil et al. 2017) constructed an Arabizi corpus and developed a character-based Arabizi to Arabic neural transliteration model. Their research demonstrates the superiority of Neural Machine Transliteration over Statistical Machine Transliteration when dealing with large corpora. The methods they employed, especially Neural Machine Transliteration, could prove useful for reverse transliterating of large amounts of dialectal Arabic data into LDA.

(Shazal et al. 2020) presented a unified model for Arabizi detection and transliteration into a code-mixed output with consistent Arabic spelling conventions, using a sequence-to-sequence deep learning model. their best system achieved 80.6% word accuracy and 58.7% BLEU score. This work, particularly their approach and achieved metrics, could provide valuable insights and benchmarks for our work on machine translation of Latinised dialectal Arabic.

3.2.2 Translation

(May et al. 2014) proposed a statistical model that used a machine translation system trained on non-Arabizi and a weighted finite state transducer-based Arabizi to Arabic conversion module. The resulting Arabic text was then translated to English. This approach could be

instructive for our work, especially regarding the conversion of LDA into MSA or dialectal Arabic before translating into another language.

(Wees et al. 2016) presented a new approach to improve the Arabizi to English translation by first transliterating Arabizi to Arabic using character mapping then utilizing phrase-based Statistical Machine Translation to translate into English. This method is significant, as it proposes a two-stage translation process, which may be an effective strategy for translating LDA.

4 Classification of Latinised Dialectal Arabic

In this chapter, we present our systematic approach to tackling the first research question. We detail the methodology and steps we plan to use for the classification and comparative analysis of these Latinised Arabic dialects.

In this chapter, we describe the systematic classification of Arabic dialects, focusing on those inscribed in the Latin script. The chapter is divided into two parts, each discussing a crucial aspect of our study.

The first part of this chapter explores the method used to gather and prepare our monolingual dialectal corpora. We discuss the process for collecting this valuable resource and the necessary steps involved in its preprocessing. The main goal is to give the reader a solid understanding of the practical approaches used in our data collection and preprocessing phases.

The second half of the chapter focuses on the classification model we used. Here, we do not just outline our chosen methodology, but dig into the reasons for selecting this particular model over others. The intent is to offer a robust understanding of the model's advantages, its suitability for our dataset, and the considerations that influenced our choice.

4.1 Data Collection

As previously mentioned, there is a scarcity of available parallel Latinised Dialect Arabic datasets. To the best of our knowledge, only two parallel LDA corpora exist: (Tracey et al. 2021)[LDA_{eg}], which comprises approximately 150,000 LDA-EN sentence pairs, and (Chen et al. 2017) [LDC2017T07], which contains around 67,000 sentence pairs between LDA and Arabic. It is important to note that both of these corpora primarily focus on the Egyptian dialect.

However, for our LDA dialect classification task, it is essential to gather LDA sentences from diverse regions across the Arab world in order to accurately distinguish between dialects. Consequently, we took on the task of collecting and constructing our own corpora to better conduct the dialect classification task.

4.1.1 Own Corpus Collection

To collect the corpora, we used the Twitter API and utilized Tweepy[1], a Python library for accessing the Twitter API. The collection process was guided by geographical information,

[1] `https://github.com/tweepy/tweepy`

in which we specified the coordinates of various regions within each country for querying purposes. Furthermore, in our query, we excluded tweets written in Arabic script to ensure a focus on LDA. Additionally, we took into account the languages of non-Arabic speaking neighboring countries of each country's query and excluded those languages from the collected tweets. This ensured a more precise and relevant dataset for our analysis. The collected tweets were subsequently extracted and labeled according to their respective countries. As shown in Listing 4.1, we provide an example of our API query for Saudi Arabia.

Listing 4.1: example of our API query for Saudi Arabia

```
(place_country:SA
OR point_radius:[46.70818 24.597204 24mi]
OR point_radius:[39.611799 24.465913 24mi]
OR point_radius:[36.539045 28.400536 20mi]
OR point_radius:[39.447662 21.496455 24mi]
OR point_radius:[39.817625 21.426172 17mi])
(-lang:ar -lang:fr -lang:en -has:links -lang:es
-lang:tr -lang:hi -lang:bn -lang:ur -lang:pa -lang:id)
```

Table 4.1: Data Collection Statistics

Country	All tweets	LDA tweets
Algeria	1,004,109	186,709
Egypt	145,294	58,016
Jordan	803,888	205,374
KSA	930,431	132,749
Lebanon	109,324	69,663
Morocco	594,162	295,180
Tunisia	-	72,929
Kuwait	-	5,037
Total	3,542,208	1,025,702

4.1.2 Preprocessing of Collected Data

After gathering a substantial amount of data from diverse regions in the Arab world, a thorough examination revealed the presence of considerable noise within the dataset. Despite the exclusion of images and videos in our query for tweet retrieval, the collected tweets remains filled with unnecessary elements, such as URLs, hashtags, Twitter mentions, and emojis. Moreover, as mentioned in the background section on Latinised Dialectal Arabic, the language itself is inherently noisy due to its primary usage on social media platforms. Social media users often employ creative spellings of words, further contributing to the linguistic noise. This phenomenon is particularly pronounced in non-standardized

languages like LDA, where individuals manipulate word lengths and abbreviations to emphasize sentiments and express emotions.

Consequently, preprocessing the dataset has critical importance as it serves two primary purposes. Firstly, it enhances the overall quality of the data, ensuring a more accurate representation of the Arabizi dialects. Secondly, it enables the development of a robust classification model tailored to the distinctive features of Arabizi. By systematically removing irrelevant elements, such as URLs and emojis, and addressing the characteristic of the language, we can refine the dataset and create a foundation for an improved classification model. This endeavor aligns with our objective of comprehensively analyzing and categorizing the diverse Arabizi dialects prevalent across the Arab world.

Our preprocessing approach consists of two main components:

Foreign Languages Elimination:

Many Arab countries share borders with non-Arabic speaking nations. For instance, in North African corpora, we encountered numerous tweets written in South European languages. Although our query (4.1) excluded neighboring languages and those spoken primarily by expatriates in a given country, some tweets from excluded languages managed to evade detection. This occurrence could be attributed to Twitter's language detection feature. To address this, we employed a Python library called Polyglot[2]. For each retrieved tweet, we utilized Polyglot to detect its language. If Polyglot reliably identified the tweet as being in any language, we removed it from our dataset since Polyglot cannot detect Arabizi.

Noise Elimination & Normalization:

Our initial step involved cleaning the acquired data from various forms of noise. This encompassed removing special characters, symbols, hashtags, mentions, and URLs, which are prevalent on Twitter. By eliminating these elements, we aimed to focus solely on the linguistic content and minimize extraneous distractions.

Additionally, we undertook the task of cleaning the textual data of emojis. Emojis are commonly used on Twitter and social media platforms in general. However, since sentiment analysis is irrelevant to our objectives, we made the decision to completely eliminate emojis from our dataset.

Furthermore, we implemented normalization techniques for Arabizi words. In certain instances, users tend to exaggerate their emotions by repeating the same letter multiple times within a word. For example, in the Arabizi word "winaaaak" (meaning "where are you" in English), the letter "a" is repeated several times unnecessarily. To address this phenomenon, we adopted a mitigation strategy by removing any repeated letter beyond the second repetition.

By employing these preprocessing techniques, we aimed to enhance the quality of our dataset by eliminating noise, and irrelevant content. This rigorous approach ensures a more accurate and focused analysis of the Arabizi dialects we are investigating.

[2] `https://github.com/aboSamoor/polyglot`

4.2 Classification of Dialects

4.2.1 The Selection of BERT

In recent research conducted by (González-Carvajal and Garrido-Merchán 2020), a comparison was made between BERT text classification and traditional NLP approaches, these traditional approaches include different popular machine learning models such as SVC or Logistic Regression that use a vocabulary extracted from a TF-IDF model. TF-IDF commonly used in traditional NLP, assigns weight to words based on their relevance and uniqueness in a document within a corpus. The findings demonstrated that BERT surpasses the traditional NLP approach in terms of performance.

For the specific task at hand, the selection of a BERT pretrained model was crucial. In a scientific review called "BERT Models for Arabic Text Classification: A Systematic Review" (Alammary 2022), it was observed that Arabic BERT models consistently exhibited superior performance in classifying Arabic text when compared to other machine-learning models.

Our research intends to carry out a comprehensive comparison between two distinct models: the Multilingual BERT and the monolingual Arabic BERT model. The Multilingual BERT model, designed to support an extensive array of languages including Arabic, may not necessarily outperform the monolingual Arabic BERT model in specific tasks, particularly Arabic text classification. This hypothesis is derived from a detailed study conducted by (Virtanen et al. 2019), wherein it was suggested that monolingual BERT models could potentially offer superior performance over the multilingual variant. However, it's crucial to note that this study primarily focused on the Finnish language. Given this language-specific perspective, the performance dynamics might differ when considering Arabic, warranting a rigorous comparison of the two models. Therefore, to gain a better understanding, we feel a comparison of these two models is imperative.

The Multilingual BERT model that we chose is "bert-base-multilingual-cased" model. This model is uniquely designed to process text from multiple languages, including Arabic, while maintaining case sensitivity to distinguish between uppercase and lowercase letters. This model follows the general BERT architecture, composed of a multi-layer bidirectional Transformer encoder. Specifically, the structure comprises 12 Transformer blocks or layers, each with a hidden size or the dimensionality of the representation of 768. Each of these layers utilizes 12 self-attention heads. The "bert-base-multilingual-cased" model encapsulates approximately 110 million trainable parameters.

4.2.2 The Selection of Arabic BERT Model

In light of this evidence, a decision was made to utilize a monolingual BERT model specifically designed for Arabic. Several high-performing Arabic BERT models exist e.g AraBERT, MARBERT, QARiB and Arabic ALBERT, differing primarily in the size and context of their training data. While some models are trained exclusively on Modern Standard Arabic (MSA), others incorporate Dialectal Arabic, and a few contain both MSA and dialectal Arabic. Our choice of model was motivated by the need for inclusivity, aiming to train a model on a dataset similar to the one used for fine-tuning.

With that taken consideration, we opted for [AraBERT-V02-Tweets] (Antoun et al. 2020). This model was pretrained on a Twitter dataset similar to our intended fine-tuning data. AraBERT-V02-Tweets is specifically tailored for the Arabic language, employing a pretraining dataset consisting of MSA from Arabic media and dialectal Arabic from Twitter. This version of AraBERT was trained on a substantial dataset comprising 200 million MSA sentences, which is 44 times larger than the Arabic pretraining dataset used for Multilingual BERT. Additionally, it underwent continued training on 60 million tweets. AraBERT-V02 has 12 transformer blocks, each containing 768 hidden units, 12 self-attention heads, and a total of 110 million trainable parameters (Antoun et al. 2020).

By selecting AraBERTv0.2-Twitter, we ensure that our model aligns closely with our fine-tuning dataset, offering optimal performance in the classification of dialects. The model's extensive training on both MSA and dialectal Arabic, coupled with its large-scale pretraining on relevant Arabic language sources, equips it with the necessary linguistic context and knowledge to effectively classify dialectal variations.

5 Machine Translation for Latinised Dialect Arabic

This chapter presents the second contribution of our thesis, which examines the effectiveness of different approaches to neural machine translation of LDA. Our thesis encompasses various stages, starting from data collection and preprocessing to dialect classification. we delve into two interesting scenarios within the landscape of machine translation for Latinised Dialectal Arabic.

The first scenario is of particular interest because it emerges from an environment where parallel data is available but remains significantly under-resourced. This situation is interesting, as it provides a platform to explore and create strategies that can extract maximum utility from minimal resources. Under such constraints, optimizing machine translation techniques poses a unique challenge and opens up a rich field for study and innovation.

The second scenario draws our attention as it presents an entirely different challenge, a total lack of parallel data. Such a situation is not uncommon when working with less-documented or minority languages. It offers a unique opportunity to investigate and apply alternative machine translation methods that can function effectively even without the typical prerequisite of parallel data.

Following these scenarios, we propose two approaches for machine translation.

In the context of the first scenario, where we are constrained by limited parallel data, we opt for a supervised machine translation approach. This approach improves the limited parallel datasets and complements them with auxiliary data. Specifically, we incorporate a transliterated version of dialectal Arabic during the training process. This allows us to enrich the resource-poor parallel data and potentially improve the quality of the output translation.

As for the second scenario, where there is no parallel data available, we utilize modified version of unsupervised machine translation by building upon a pretrained cross-lingual translation model and auxiliary data. UNMT learns to translate without the need for parallel data, only by training on monolingual data. UNMT is particularly well-suited to this task as it can generate translations without requiring parallel data, making it an ideal strategy for such resource deprived situations.

By systematically evaluating these scenarios and techniques, our research aims to enhance translation quality and effectively navigate the challenges associated with under-resourced or resource-deprived LDA.

5.1 Supervised Neural Machine Translation with Auxiliary Data

In this section, we address our second research question (RQ2): "How can we enhance the translation quality from Latinised Dialectal Arabic into English in scenarios where parallel data resources are limited?" We will explain the methodologies and approaches employed to tackle this challenge.

5.1.1 Preprocessing Arabic Script as Auxiliary Data

The aim of this experiment is to strengthen the performance of machine translation models through the incorporation of preprocessed Arabic script as auxiliary data.

In this research we test three preprocessing pipelines. The motivation for exploring multiple preprocessing methods in this research is rooted in our aspiration to extract the maximal possible informational value from the Arabic script. The Arabic language, with its rich morphological structure and complex dialectal variations, holds a wealth of linguistic information. By applying diverse preprocessing methods, we strive to identify and harness the optimal technique that can fully unlock this potential. This involves carefully extracting, preserving, and utilizing the inherent structural and semantic elements of the language during the preprocessing phase. The ultimate aim to find the best preprocessing method, that our machine translation models can most effectively use, thereby enabling the production of more accurate translations. We applied a range of methodologies in our

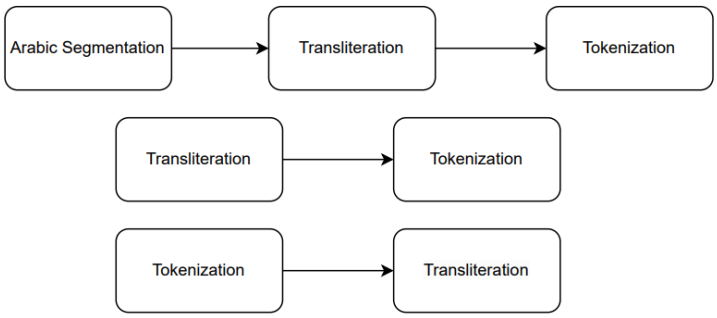

Figure 5.1: Pre-processing techniques evaluated for enhancing translation quality

exploration. One such approach involved using Farasa, a state-of-the-art Arabic text processing toolkit, for segmentation. Developed by (Abdelali et al. 2016), Farasa ('insight' in Arabic) offers a set of tools for fundamental Natural Language Processing tasks, including text segmentation, part-of-speech tagging, and named entity recognition, among others. For our purposes, the Farasa segmenter is for our case interesting, with its capability to break down Arabic words into their individual components, namely, the prefix, stem, and suffix. This segmentation is invaluable given the rich morphology of Arabic, where a single word can often carry the meaning equivalent to a full sentence in English.

Following the segmentation with Farasa, the text underwent transliteration using the Buckwalter system and was subsequently tokenized. This method was compared with two

others: one involving only transliteration and tokenization, and another implementing tokenization prior to transliteration.

Figure (5.1) presents an overview of the pre-processing techniques we assessed with the objective of enhancing the auxiliary data deployed in our machine translation models. Furthermore, Figure (5.2) provides an exposition of our selected preprocessing pipeline. This commences with word segmentation via Farasa, followed by a sentence-level transliteration using the Buckwalter encoding system. The final step in our pipeline is the tokenization of the transliterated text, effectively readying it for assimilation into our translation model.

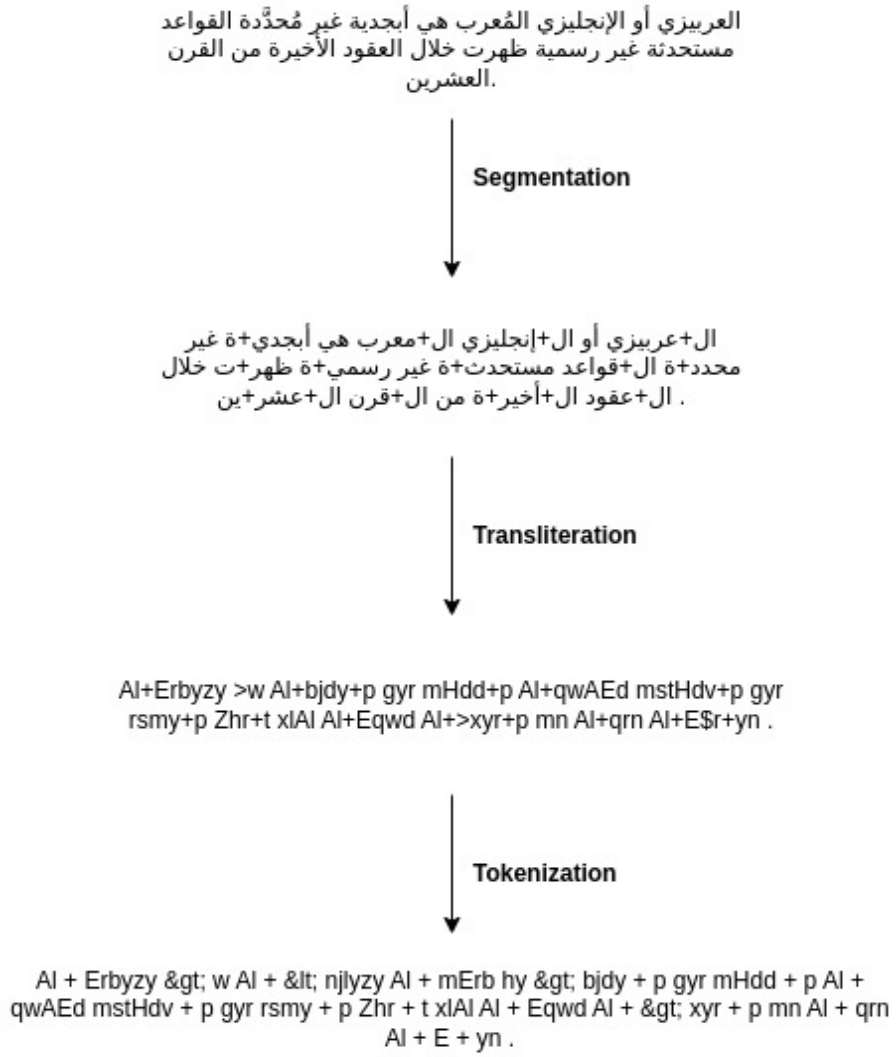

Figure 5.2: The Arabic sentences pre-processing pipeline

5.1.2 Utilization of Arabic Script Transliterations as Auxiliary Data for LDA Translation

Latinised Dialectal Arabic, considered a low-resource language, is burdened with the limitation of small parallel corpora. This scarcity poses a critical challenge to language modeling and machine translation tasks. Nevertheless, this issue can be potentially mitigated by leveraging the intrinsic linguistic relations between LDA and its closely-related languages in particular its Arabic script counterpart, Dialectal Arabic and MSA.

Given the substantial overlap in linguistic features between DA and LDA, exploiting DA resources can considerably augment the limited LDA datasets. By mapping the Arabic script from DA into the Latin script of LDA through transliteration, we can generate new training instances that potentially enrich the LDA corpus, hence facilitating more robust language models.

Moreover, DA shares linguistic similarities with Modern Standard Arabic, enabling further augmentation possibilities. MSA, widely recognized for its comprehensive coverage of Arabic grammar and vocabulary, presents an opportune auxiliary data source to enlarge the LDA dataset. Through a similar process of transliteration, the MSA corpus can be transformed into additional LDA instances, hence significantly increasing the size of the LDA corpus.

The paper (Chakravarthi et al. 2019) compared the effect of utilizing of training data from closely-related transliterated languages on under-resourced Dravidian languages and showed the benefits of transliteration in supervised Multilingual Neural Machine Translation. Motivated by these findings, we integrate transliteration into our supervised NMT model with the aim of boosting its performance in LDA translation tasks.

Utilization of Transliterated Dialectal Arabic for LDA Translation Quality

To tackle this problem and to enlarge our training dataset, we adopt a strategy of transliteration. Given the abundant resources of DA, we employ transliteration of DA script into Latin characters. This process results in a 'Latinised' variant of DA, effectively bridging the gap between DA and LDA in terms of scriptural representation and significantly expanding the scope of 'Latinised' training data.

we train our supervised NMT model on transliterated DA data alongside the existing LDA data. This practice is expected to introduce a richer dialectal context to the model, thereby enhancing its ability to grasp the characteristics of LDA. The model's exposure to a broader spectrum of linguistic variants may contribute to a better understanding of LDA's nuanced dialectal complexities. Therefore the enhanced dataset, born out of the transliteration of DA, forms a strategic addition to the learning mechanisms of our model.

Utilization of Transliterated MSA for LDA Translation Quality

The transliteration of Modern Standard Arabic into Latin characters could have impact on the translation quality of LDA. The auxiliary data derived from the transliteration process contributes to the supervised and learning mechanisms of the model, potentially enabling

the model to learn subtle linguistic patterns and relationships across LDA and other similar languages involved.

5.2 Unsupervised Machine Translation with Multilingual Transfer

In this section, we tackle our third research question (RQ3): "How can we enhance the translation quality from Latinised Dialectal Arabic into English in the absence of parallel data resources?" We will discuss the methodologies and approaches utilized to overcome this challenge without relying on any parallel data resources between LDA and any other language.

As previously mentioned LDA is a language variant characterized by a low-resourced in terms of parallel datasets. Traditional machine translation methods tend to underperform in such contexts, emphasizing the need for innovative approaches to address this gap. This thesis utilizes building upon a multilingual pre-trained model, a strategy proven to build superior model generalization across diverse linguistic scenarios (Garcia et al. 2020; Koneru et al. 2021; Li et al. 2020; Liu et al. 2020)

5.2.1 Transliteration

Beyond its role in supervised learning, in paper (Koneru et al. 2021) the auxiliary data derived from the transliteration process also contributes to the Multilingual Unsupervised Machine Translation (MUNMT) component of our model. (Moosa et al. 2023) demonstrates that the act of transliterating closely related languages into a common script markedly enhances performance, especially in comparatively low-resource languages. This improvement not only boosts multilingual language model performance but also fosters the development of more robust cross-lingual representations.

Here, the transliterated data, which includes Dialectal Arabic, is expected to enrich the unsupervised learning mechanism of our model. This could potentially empower the model to discern and learn subtle linguistic patterns and relationships across other languages involved. Detailed insight into the utilization of this transliterated data, particularly on how it will be integrated into the model and contribute to the MUNMT, will be provided in the following section.

The preprocessing pipeline - including transliteration - is similar to the one mentioned in the previous section (5.2). Where we used Farasa, a comprehensive Arabic text processing toolkit, for segmenting the Arabic words into prefix, stem, and suffix components, which is crucial given the rich morphology of Arabic. After segmentation, the text was transliterated using the Buckwalter system and then tokenized.

5.2.2 Utilization of DA in Unsupervised MT

Initially we pretrain a cross-lingual language model (XLM) by training the model on monolingual corpora from multiple languages relevant to the system. The languages include LDA, transliterated DA, and English. This process employs the technique of Masked

Language Modeling. (Lample and Conneau 2019; Lample, Ott, et al. 2018) shows that pretraining with the MLM objective provides better initialization of supervised and unsupervised neural machine translation systems, which our experiment Machine Translation for Latinised Dialect Arabic will be focusing on.

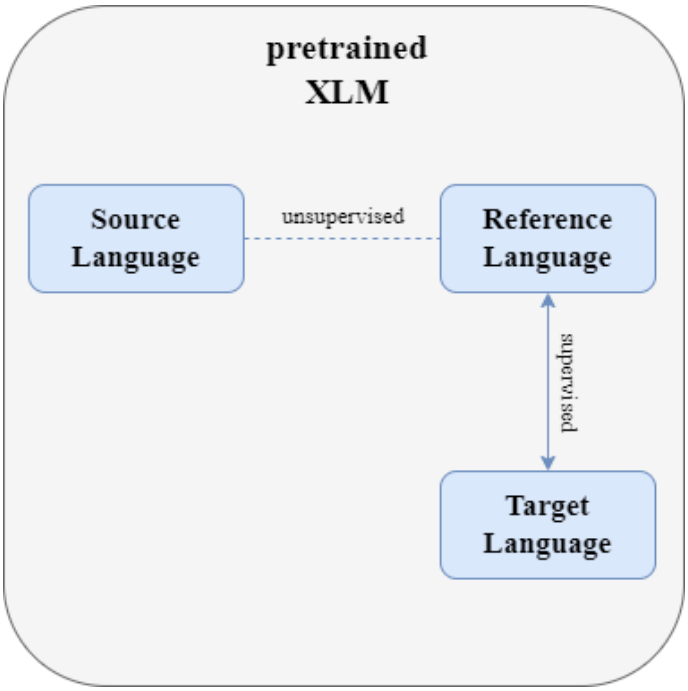

Figure 5.3: Visual Representation of the Multilingual Unsupervised Machine Translation (MUNMT) Framework.

Expanding on our main goal of translating from LDA to English, it is essential to delve deeper into the process of fine-tuning the pretrained cross-lingual model. This process, as visually depicted in Figure 5.3, necessitates two key steps: supervised and unsupervised training.

The first step involves supervised training, which acts as a bridge between the reference language - in this case, transliterated dialectal Arabic - and English. During this phase, the model is presented with parallel sentence pairs from these two languages. These pairs enable the model to identify the semantic and syntactic mappings between them, thereby equipping it with the necessary skills to generate accurate translations.

The second step entails unsupervised translation between LDA and the reference language. Unlike the previous step, this phase does not rely on parallel sentence pairs. Instead, it utilizes back-translation and Auto Encoding combined in the equation (2.1). The unsupervised translation process guided by this equation, a formula that encapsulates the principles of unsupervised learning. It weighs and optimizes the loss function of the model, ensuring that the model minimizes errors and improves its translation accuracy over time.

6 Experiments and Evaluation

This chapter embarks on an examination of the conducted experiments, offering a detailed evaluation and discussion of their outcomes.

In the first section, we address the experiment involving dialect identification. This section details the preparation of data and the fine-tuning of the BERT models employed. Subsequently, we present the results, followed by a comprehensive discussion.

The second section delves into the realm of supervised Machine Translation experiment. This context implies the existence of parallel data between LDA and English. We explain the process of corpus preprocessing, demonstrate multiple experiments conducted on diverse datasets, and subsequently discuss and compare the performance of the resulting translation models.

In the final section, we navigate through the scenario lacking parallel data between LDA and English. Here, we employ an intermediate LDA-similar reference language with the aim of enhancing the accuracy of translation. The model is trained in an unsupervised approach, utilizing a pre-trained multilingual language model. This section outlines the steps involved in data preparation for the experiments, explains the experiment settings, presents the results, and subsequently offers a discussion evaluating these outcomes.

6.1 Dialect Identification

As discussed in section 4.2 (Classification of Dialects), our choice of employing specific BERT Models is driven by a variety of factors. For our monolingual BERT model have opted for huggingface's 'aubmindlab/bert-base-arabertv02-twitter' model for optimal dialect identification, with further details available in Section (4.2). Alongside this, we have also decided to utilize the Multilingual BERT model, 'bert-base-multilingual-cased' to be specific. This choice enables us to draw a comprehensive comparison between these monolingual and multilingual BERT models.

6.1.1 Data Preparation and Encoding

To effectively fine-tune the BERT Model, we employed a carefully selected subset of the dataset delineated in Section 4.1.1 (Own Corpus Collection). This data, originally extracted from Twitter, was curated to ensure a balanced representation of varying Arabic dialects.

The selection process encompassed certain filtering steps to maintain the dataset's quality. Firstly, entries containing fewer than four words were excluded, given their inability to provide sufficient, reliable linguistic elements for successful dialect identification. Secondly, rows with fewer than three unique characters were removed. This step was taken to mitigate the presence of data points primarily composed of emotional expressions,

such as 'hahahaha' and 'lol', which offer limited value in distinguishing dialects. Lastly, duplicate entries were purged to eliminate redundancy and potential bias in the data.

Following the filtering process, each sentence in the refined dataset was prepared for further processing using the code in Listing 8.1. This involved tokenization and encoding using the BERT tokenizer. Each sentence was fed into the tokenizer's function, where several operations were performed.

These operations included the addition of special tokens '[CLS]' and '[SEP]' at the beginning and end of each sentence, respectively. The sentences were truncated and padded to a maximum length of 64 tokens to ensure uniformity in sentence length. The tokenizer's output was then converted into PyTorch tensors for compatibility with the BERT model.

To assist the model in differentiating actual content from padding, attention masks were also created. These masks enable the model to focus only on the relevant tokens in each sentence. As the encoding function outputted the encoded sentence and its corresponding attention mask, both were collected and stored for the upcoming model training phase.

The dataset includes a diverse distribution of sentences from various Arabic-speaking countries. The distribution is presented in Table 6.1:

Country	**Nr sentences**
Algeria	11664
Egypt	23744
Jordan	11624
Gulf	13082
Lebanon	27894
Morocco	6654
Tunisia	43935
All	138597

Table 6.1: Distribution of Training Sentences by Geographic Origin

6.1.2 BERT Fine-tuning

When fine-tuning our model, we adhered to the hyperparameter values suggested by the BERT paper (Devlin et al. 2018) as optimal. These values are as follows:

- Batch size: 16, **32**
- Learning rate (Adam): 5e-5, 3e-5, **2e-5**
- Number of epochs: 2, 3, **4**

The provided fine-tuning training implementation in Listing 8.2 using PyTorch illustrates a training loop for fine-tuning a pre-trained BERT model over 4 epochs. For each epoch, the model iterates over the training data in batches. In each iteration, the model resets the gradients, performs a forward pass using the batch data, computes the loss, and

backpropagates this loss to calculate gradients. The gradients are then clipped to prevent explosion and the model parameters are updated using the computed gradients through the optimizer. The learning rate is also adjusted in each step as per the scheduler. Finally, the average training loss for the epoch is calculated.

6.1.3 Experimental Results and Analysis

Model	Accuracy	Macro Avg F1	Weighted Avg F1	Sentences
AraBERT	0.85	0.82	0.85	34650
MultiLingual BERT	0.87	0.83	0.87	34650

Table 6.2: Models Overall Classification Metrics on Test Dataset

Overall Performance Comparison

Our comparative analysis of the AraBERT and Multilingual BERT classification models for Latinised Arabic dialects reveals interesting insights (see 6.1.3). Both models were evaluated on their performance over 7 dialects and the results were notably close.

The AraBERT model demonstrated a commendable performance with an accuracy score of 85%, indicating its proficiency in correctly classifying the Latinised dialectal Arabic sentences on the test dataset. This achievement is particularly significant considering the linguistic intricacies associated with different Arabic dialects. On the other hand, the Multilingual BERT model showed slightly superior performance with an accuracy score of 87%, pointing to its capability to handle the linguistic subtleties with slightly more precision.

In terms of the macro and weighted averages, AraBERT posted F1 scores of 0.82 and 0.85 respectively, implying a consistent and reliable performance across the dialects. Multilingual BERT, however, outperformed slightly with scores of 0.83 and 0.87 respectively. The macro average gives equal weight to each class, while the weighted average considers the size of each class. The slightly higher scores of Multilingual BERT in both these metrics indicate its slight edge in handling larger class sizes and maintaining balance across classes.

Impact of Data Size on Classification Accuracy

Upon analyzing the performance of our dialect classifier model, several noteworthy patterns and tendencies emerge. Most prominently, it is evident that the Tunisian dialect, which has the most extensive training dataset of 43,935 (see Table 6.1) sentences, is classified with the highest accuracy of 96% in both models. This is followed by the Lebanese and Egyptian dialect, both classified with 84% accuracy with the AraBERT model and 86% and 85% with Multilingual BERT,respectively. These dialects have dataset of 27,894 and 23,744 training sentences respectively. Such a pattern strongly suggests a positive correlation between the size of the training datasets and the classification accuracy.

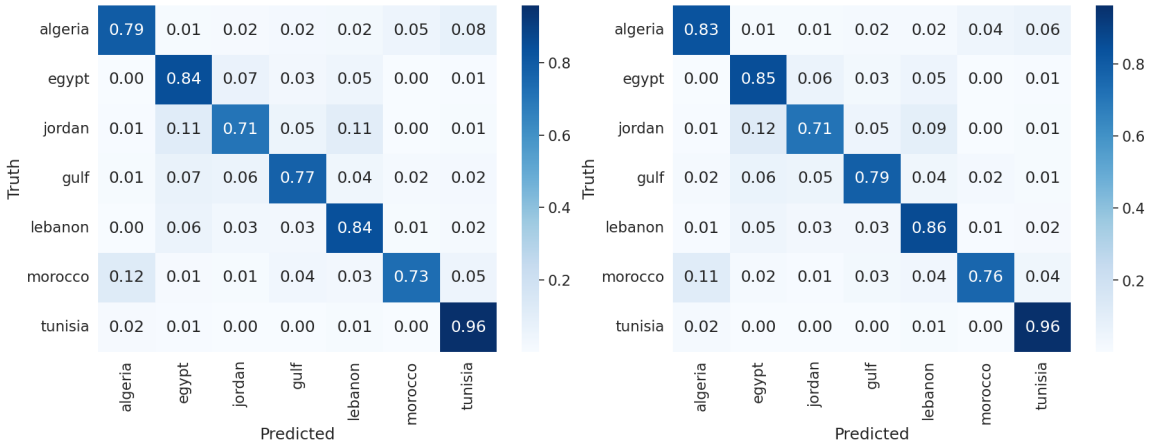

Figure 6.1: On the left ARABERT Arabic Dialects Classification Heatmap and on the right Multilingual BERT Arabic Dialects Classification Heatmap

Analysis of Dialectic Confusion and Dialect Similarity

An intriguing pattern of dialectic confusion is observable among North African dialects, which are frequently misidentified as each other. The Algerian dialect, in AraBERT for instance, is classified correctly 79% of the time but gets mistaken as Moroccan and Tunisian 5% and 8% of the time, respectively (see 6.1). This misclassification could be attributable to the geographical proximity of these countries and their shared linguistic elements, such as the prevalent use of French words in their everyday speech.

This phenomenon is not unique to North African dialects. Observations for the Egyptian and Jordanian dialects show a similar pattern. The AraBERT classification of Egyptian dialect is labeled correctly 84% of the time but is mistaken as Jordanian and Lebanese 7% and 5% of the time, respectively. Similarly, the Jordanian dialect is identified correctly 71% of the time, with an 11% chance of being misidentified as either Egyptian or Lebanese. This data implies shared linguistic features, influenced by geographical adjacency.

Among these, the most significant classification errors occur with the Moroccan dialect. Despite being classified correctly 73% of the time, when using AraBERT ,it's misidentified as Algerian 12% of the time and Tunisian 5% of the time. These frequent misclassifications may be due to the relative size of the Moroccan training dataset of only 6,654 examples and the considerable overlap in vocabulary between the Moroccan and Algerian dialects. Additionally, this misclassification might be heightened by the common use of French words within both dialects.

On the other hand, the Tunisian dialect exhibits the highest classification accuracy at 96%, with only a minimal 2% chance of being misidentified as Algerian. This high degree of accuracy might be more attributable to the larger size of the Tunisian training dataset.

6.2 Supervised Machine Translation

The aim of these experiments is to investigate the methodologies for efficient translation of LDA, particularly in scenarios where parallel data is limited. A comprehensive explanation

of this approach can be found in 5.1 (Supervised Neural Machine Translation with Auxiliary Data), wherein we delve into the enhancing of translation in settings constrained by scarce resources. This exploration is intended to identify effective strategies to optimize the translation process and to ensure accurate and coherent output, even under conditions of data limitation. This experiment is mainly exploring RQ2.

6.2.1 Data Preperation

In the case of Dialectal Arabic sentences—as illustrated in Figure (6.2)—we first needed to engage in the process of sentence segmentation. To accomplish this, we employed the Farasa tool (Abdelali et al. 2016). Following this segmentation, we then begin on the task of transliterating the segmented Arabic sentences using Buckwalter transliteration. Consequently we utilized the Moses tokenizer, which effectively broke down the transliterated sentences into a structured set of individual tokens suitable for further analysis.

Contrarily, for both English and LDA sentences, our approach was simpler. We only subjected these sentences to the tokenization process, employing the Moses tokenizer for this purpose.

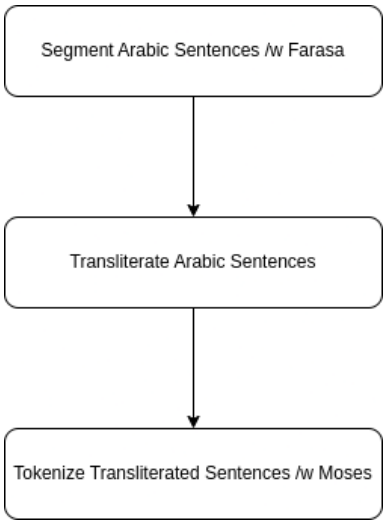

Figure 6.2: Visual representation of the preprocessing of the dialectal Arabic sentences. (Abdelali et al. 2016) was used for dialectal Arabic segmentation.

6.2.2 Architecture and Hyperparameters

After preprocessing the data according to Fairseq requirements, we will proceed to train a supervised Translation model using Fairseq. The selected architecture for this task is the Transformer model, which encompasses the following encoder/decoder parameters. The input word embeddings in both the encoder and decoder have a dimensionality of 512. The intermediate layer within the feed-forward network of the encoder/decoder has a dimensionality of 1024. Each encoder/decoder employs a multi-head attention mechanism

with 4 attention heads. The model consists of 6 layers in total, contributing to its depth and capacity for capturing complex dependencies.

Furthermore, the decoder in this model shares its embedding layer between the input (source) and output (target) tokens, meaning that the same embedding layer handles the conversion of both types of tokens into their corresponding vector representations during the decoding process. This sharing of parameters promotes parameter efficiency and allows the model to leverage the relationships between the source and target tokens for improved translation quality.

During training, the model utilizes the ADAM optimizer with beta values of 0.9 and 0.98. The learning rate is set to 5e-4, ensuring an appropriate step size for updating the model parameters. The learning rate scheduler follows the inverse square root function, adapting the learning rate based on the number of updates. Additionally, a dropout rate of 0.3 is applied as a regularization technique to prevent overfitting.

6.2.3 Experiments

The foundation of our first experiment lies in the deployment of a Modern Standard Arabic dataset ($MSA_{Opensub}$), specifically the corpus of opensubtitles. To maintain comparative consistency across all experiments, we have selected 250,000 sentences from a broader dataset, ensuring relative equality in datasets size.

The second experiment focuses on the LDC2021T15 (LDA_{eg}) dataset, comprising exclusively of Egyptian LDA parallel corpus. This selection displays the performance of our translation model when trained with a dataset focused only on LDA content.

Our third experiment utilizes with the LDC2012T09 (DA_{eg_lev}) dataset, which constitutes sentences of Egyptian and Levantine Dialectal Arabic sentences. These sentences underwent a process of transliteration.

In the last experiment, we combined the datasets DA_{eg_lev} and LDA_{eg} ($DA_{eg\_lev}\_LDA_{eg}$). The underlying goal was to assess the potential of transliterated dialectal Arabic, as auxiliary data, in enhancing the translation quality from LDA into English.

6.2.4 Experimental Results and Analysis

test_data \ training_data	$MSA_{Opensub}$	LDA_{eg}	DA_{eg_lev}	$DA_{eg\_lev}\_LDA_{eg}$
LDA	0.54	21.69	0.00	**29.12**
Egyptian	5.26	0.43	28.87	**32.31**
Jordanian	4.06	0.48	19.88	**21.05**
MSA	11.10	0.33	18.89	**21.02**
Palestinian	3.83	0.45	17.50	**18.23**
Syrian	5.00	0.50	20.00	**21.78**
Tunisian	3.72	0.34	12.33	**13.77**
Opensub(MSA)	**27.51**	0.35	7.71	10.55
Average	7.57	3,07	15.65	**20,98**

Table 6.3: Supervised Machine Translation's Experiments BLEU scores

Preprocessing Effect on Translation Quality

As mentioned in the previous chapter (5.1.1). We experimented with multiple preprocessing approaches on the dataset DA_{eg_lev}. One approach involved segmentation with Farasa, followed by Buckwalter transliteration and tokenization, which resulted in a BLEU score of 21.78 on the test dataset. Another approach involved only transliteration and tokenization, resulting in a BLEU Machine Translation for Latinised Dialect Arabic score of 20.31. Finally, tokenization followed by transliteration resulted in a BLEU score of 20.96. We ultimately chose to use the first approach for all of our experiments, which was segmentation with Farasa followed by transliteration and tokenization.

Effect of MSA on LDA Translation

As we can observe in the experiment $MSA_{Opensub}$, the supervised machine translation model trained on transliterated Modern Standard Arabic extracted from Opensub dataset notably performed best on the MSA dataset, achieving a BLEU score of 27.51. This confirms the expectation that a model trained on a particular language or dialect would perform optimally when translating content from that specific source. However, the BLEU scores were significantly lower when the model encountered dialects or LDA, indicating a difficulty in generalizing from MSA to LDA with only 0.54 BLEU score in our case.

Effect of DA on LDA Translation

In the experiment DA_{eg_lev} focused on transliterated Egyptian and Levantine Dialectal Arabic, resulting in relatively high BLEU scores for these dialects, 28.87 and 19.88 respectively, and a still relatively respectable score for MSA at 18.89. This result implies that training the model on dialectal Arabic provided some level of translational alignment with MSA. However, the score for LDA fell to zero, again underscoring the difficulty the MT model has in bridging between LDA and the other Arabic forms.

Baseline

In our thesis LDA_{eg} is the only parallel LDA corpus that we have. This dataset contains Egyptian dialect only. We utilize it to train a translation model to serve as our reference point or our baseline score for experiments. Training the NTM model on the LDA corpus, showed a noticeable improvement in the translation of LDA with a BLEU score of 21.69. This further supports the notion that a model's performance is directly linked to the specifics of its training data. However, the model's performance on non-LDA Arabic dialects and MSA was extremely poor, suggesting a considerable gap between LDA and transliterated Arabic script and MSA, at least as captured by the MT model with this transliteration method.

Effect of DA as Auxiliary Data on LDA's Translations Quality

In $DA_{eg\_lev}\_LDA_{eg}$, the gains witnessed in this experiment are substantial and offer promising prospects for further application of this approach. Focusing on LDA, the inclusion into the dialectal Arabic dataset led to a remarkable increase in BLEU scores - from 21.69 to 29.12, indicating an increase of over 7 BLEU points. This is a significant improvement, demonstrating the value of integrating transliterated dialectal Arabic as auxiliary data to refine the translation quality of LDA.

In terms of specific dialects, the Egyptian dialect registered noteworthy gains as well. The BLEU score increased from 28.87 to 32.31, which represents a leap of more than 3 BLEU points. The significant gain observed in the Egyptian dialect, more so than other dialects, can likely be attributed to the fact that the included LDA_{eg} predominantly contains the Egyptian dialect. This dominance means that there was a significantly larger corpus of Egyptian dialect data available for training, which in turn led to better performance when the model was tested on the Egyptian dialect.

Furthermore, this integration of LDA into the dialectal Arabic dataset has evidently uplifted the translation quality across all dialects, including MSA, suggesting a mutual improvement effect. This implies that dialectal and LDA data can complement each other in a training context.

Overall performance of the translation models

The results of these five experiments reveal the importance of dataset diversity and size in supervised machine translation. The low average BLEU scores in the first two experiments highlight the challenges of models trained only on Modern Standard Arabic or LDA. These models struggled due to significant linguistic differences among dialects. However, experiments three and four showed improved results. Using dialectal Arabic datasets, these experiments yielded higher average BLEU scores of 15.65 and 15.86. This shows the benefit of using diverse dialectal data and increasing the dataset size. The last experiment showed the most significant increase. By combining dialectal Arabic with LDA, the average BLEU score increased to 20.98. This suggests that integrating different types of data in training can greatly enhance model performance.

6.3 Unsupervised Machine Translation with Multilingual Transfer

In addressing RQ3, we explore how to improve MT quality in cases where parallel data between LDA and English is not available. Our approach combines the reference language of transliterated dialectal Arabic with both supervised and unsupervised machine learning techniques. As previously outlined in Section 5.2, supervised training focuses on the correlation between the reference language DA and English. In contrast, unsupervised training, which includes backtranslation and autoencoding, operates between the reference language and LDA. Our strategy is based on a pretrained cross-lingual language model, providing a foundation for additional learning and fine-tuning.

6.3.1 Data Preperation

The pipeline of data preparation concerning the segmentation, transliteration and tokenization uses the same technologies and analogous to the supervised MT approach in the previous section 6.2.1 By adhering to analogous data preparation pipelines, we assure uniformity and comparability across our research.

6.3.2 Architecture and Hyperparameters

In this experiment, the system is divided in two main stages in pretraining and training stages, both utilizing a Transformer-based design. The pretraining phase applies an MLM objective for transliterated DA, LDA and English. The model adopts an embedding dimension of 1024, a six-layer structure, and eight attention heads. The Adam optimizer facilitates model optimization with a learning rate of 0.0001, while GELU activation and dropout serve to optimize the model's performance.

The training phase retains the same Transformer parameters as were applied during pretraining. A scheduled lambda auto-encoding at '0:1,100000:0.1,300000:0' is applied to lessen the auto-encoding loss impact over time. Maintaining the same Transformer parameters as in pretraining, the optimization process employs Adam with Inverse Square Root Learning Rate Schedule. The learning rate remains at 0.0001, with betas set at 0.9 and 0.98. A token per batch limit of 2000 is enforced, with a batch size of 32 and a sequence length of 256. To carry out the proposed experiments, we use the XLM[1].(Lample and Conneau 2019) toolkit by Facebook Research.

6.3.3 Experiment

In this experiment, we designate English as the target language and LDA as the source language. We employ Dialectal Arabic as an intermediate reference language to bridge these two. A visual representation of this experimental configuration is presented in Figure 6.3. The LDA corpus is extracted from the *LDC2021T15* (LDA_{eg}) dataset, which serves as a monolingual dataset. We utilize the *LDC2012T09* (DA_{eg_lev}) dataset as the basis for

[1] https://github.com/facebookresearch/XLM

the Dialectal Arabic-English parallel corpus. Notably, prior to its integration into our experiment, Dialectal Arabic undergoes a process of segmentation, transliteration, and tokenization, ensuring standardized representation and facilitating effective utilization within the multilingual translation framework.

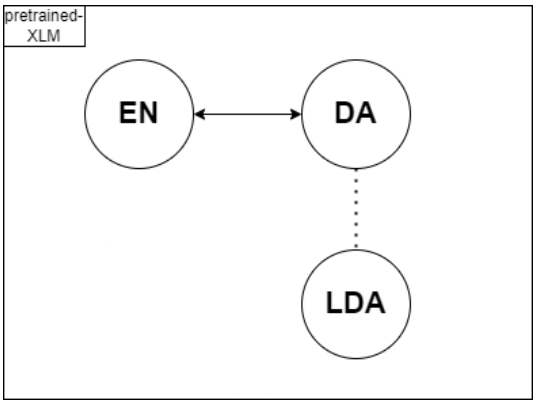

Figure 6.3: Visual Representation of Our Experiment Ref_{DA}. The dotted line represents unsupervised training and the double sided arrow represents supervised training

6.3.4 Experimental Results and Analysis

Experiment / test_data	Ref_{DA}	DA_{eg_lev}
LDA	6.68	0.00
Egyptian	29.75	28.87
Jordanian	17.46	19.88
MSA	18.28	18.89
Palestinian	15.72	17.50
Syrian	20.06	20.00
Tunisian	12.31	12.33
Average	17.18	15.65

Table 6.4: Results Comparison of Multilingual Unsupervised Machine Translation Model Ref_{DA} and supervised Machine Translation Model DA_{eg_lev} (see 6.2.3)

Overall Performance

The comparison of Ref_{DA} and DA_{eg_lev} offers interesting insights into the performance of the supervised and unsupervised machine translation systems on various Arabic dialects. Ref_{DA}, which utilizes a crosslingual language model with DA as reference language and supervised training between dialectal Arabic and English, showed superior average performance, with an overall average BLEU score of 17.18 compared to 15.65 in DA_{eg_lev}.

The Impact of Dialectal Arabic as a Reference Language on LDA

Our experiment, as shown in Table 6.4, highlights the significant effect of incorporating Dialectal Arabic as a reference language in the LDA model. Comparing the results to the supervised experiment (DA_{eg_lev}), we observe an improvement in LDA's BLEU score, achieving 6.68 compared to the previously obtained score of 0.0. It is worth noting that both experiments were trained on the same parallel dataset between DA and English.

The key distinction lies in the second experiment presented in Table 6.4, where, in addition to utilizing a pretrained crosslingual model, we also incorporate unsupervised training between LDA and DA. This additional training proved to be highly beneficial, leading to a significant boost of over 6 BLEU points in translation performance.

The Impact of the Translation System on Arabic Dialects

The application of a crosslingual language model with reference language and the incorporation of supervised training between dialectal Arabic and English combined with unsupervised training between LDA and DA in the experiment Ref_{DA} had visible impacts on the translation performance across various Arabic dialects. However the effect was not uniformly distributed among dialects. For instance, the Egyptian dialect showed the most notable performance in Ref_{DA}. A reason for this improvement only on Egyptian Dialect could be due to the fact that the LDA dataset contains solely Egyptian data. However, the Jordanian, MSA, Palestinian, and Syrian dialects had slightly better results in the purely supervised setting, suggesting that the impact of the crosslingual language model and unsupervised training may not be equally beneficial for all dialects. Furthermore, the performance on the Tunisian dialect remained almost unaffected, indicating that the benefits of unsupervised training might be limited for this specific dialect.

7 Conclusion

7.1 Work Summary

We revisit the RQs we outlined in our introduction.

RQ1:How efficiently can we distinguish LDA dialects? Our research indicated a positive correlation between the volume of the training datasets used to fine-tune BERT and the accuracy of dialect classification. Specifically, the AraBERT based classification model reached an accuracy of 85% and an F1 score of 85. However, the Multilingual BERT based classification model outperformed our monolingual BERT, achieving an accuracy of 87% and an F1 score of 87 when implemented one the same subset of our collected data from Twitter tweets. It was also observed that geographically neighboring Arab countries tend to share linguistic similarities, which often led to misclassifications. Hence, achieving a high level of accuracy in dialect classification is dependent not only on the size of the training data but also on the linguistic nuances and geographical proximity of the dialects in question.

RQ2:In the case of limited parallel data, how to improve translation quality from LDA into English? The study found that the incorporation of transliterated dialectal Arabic as auxiliary data into the machine learning algorithm had a significant positive impact on translation quality, resulting in a 7-point increase on the BLEU scale. However, the inclusion of MSA data did not yield a similar improvement. Notably, the use of Egyptian LDA was shown to enhance the translation quality of the Egyptian Arabic dialect by an additional 3 BLEU points, indicating the potential of this technique for specific dialects.

RQ3:In the case of no existing parallel data, how to improve translation quality from LDA into English? Here, the study employed an approach that utilized transliterated dialectal Arabic as a reference language, coupled with a pretrained crosslingual language model. This strategy resulted in an improvement in translation quality, as evidenced by a 6-point increase on the BLEU scale. Thus, the research concluded that the integration of dialectal Arabic as a reference language, coupled with the use of a crosslingual language model, serves as an effective strategy for enhancing translation performance when parallel data are not available.

7.2 Future Work

While this thesis has offered insights into the dimensions of Arabizi, it also highlights several potential areas for future inquiry. Firstly, the application of BERT in machine translation presents notable potential, as evidenced by encouraging results presented (Zhu et al. 2020) and also the promising results it showed in the dialect classification experiments. Secondly, there is an evident need for the accumulation and enhancement

of parallel corpora relevant to Arabizi due to the current lack thereof. This scarcity of resources remains a significant challenge to the progress of machine translation within this domain. Lastly, the potential incorporation of LDA in a multilingual context could be particularly beneficial. Such models have demonstrated effectiveness by exploiting languages abundant in data to support those languages lacking substantial data. This approach, applied to LDA, could serve to further increase the efficacy and applicability of cross-lingual language models.

Bibliography

Abdelali, Ahmed, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak (2016). "Farasa: A fast and furious segmenter for arabic". In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Demonstrations*, pp. 11–16.

Aboelezz, Mariam (2009). "Latinised Arabic and connections to bilingual ability". In: *Lancaster University Postgraduate Conference in Linguistics and Language Teaching*. Vol. 3, pp. 1–23.

Adouane, Wafia, Nasredine Semmar, Richard Johansson, and Victoria Bobicev (Dec. 2016). "Automatic Detection of Arabicized Berber and Arabic Varieties". In: *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 63–72. URL: https://aclanthology.org/W16-4809.

Alabdulqader, Ebtisam, Majdah Alshehri, Rana Almurshad, Alaa Alothman, and Noura Alhakbani (2014). "Computer-Mediated Communication: Patterns and Language Transformations of Youth in Arabic-Speaking Population". In: *Information Technology & Computer Science (IJITCS)* 17.1, p. 85.

Alammary, Ali Saleh (2022). "BERT models for Arabic text classification: a systematic review". In: *Applied Sciences* 12.11, p. 5720.

Allehaiby, Wid H (2013). "Arabizi: An Analysis of the Romanization of the Arabic Script from a Sociolinguistic Perspective." In: *Arab World English Journal* 4.3.

Antoun, Wissam, Fady Baly, and Hazem Hajj (2020). "AraBERT: Transformer-based Model for Arabic Language Understanding". In: *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, p. 9.

Al-Badrashiny, Mohamed, Ramy Eskander, Nizar Habash, and Owen Rambow (2014). "Automatic transliteration of romanized dialectal arabic". In: *Proceedings of the eighteenth conference on computational natural language learning*, pp. 30–38.

Baert, Gaétan, Souhir Gahbiche, Guillaume Gadek, and Alexandre Pauchet (2020). "Arabizi language models for sentiment analysis". In: *Proceedings of the 28th international conference on computational linguistics*, pp. 592–603.

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

Berrimia, Mohamed, Abdelouahab Moussaouia, Mourad Oussalahb, and Mohamed Saidia (2020). "Arabic dialects identification: North African dialects case study". In.

Bianchi, Robert Michael (2012). "3arabizi-When local Arabic meets global English". In: *Acta Linguistica Asiatica* 2.1, pp. 89–100.

Bies, Ann et al. (Oct. 2014). "Transliteration of Arabizi into Arabic Orthography: Developing a Parallel Annotated Arabizi-Arabic Script SMS/Chat Corpus". In: *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*. Doha, Qatar:

Association for Computational Linguistics, pp. 93–103. DOI: 10.3115/v1/W14-3612. URL: https://aclanthology.org/W14-3612.

Caswell, Isaac, Theresa Breiner, Daan van Esch, and Ankur Bapna (2020). *Language ID in the Wild: Unexpected Challenges on the Path to a Thousand-Language Web Text Corpus*. arXiv: 2010.14571 [cs.CL].

Chakravarthi, Bharathi Raja, Mihael Arcan, and John P McCrae (2019). "Comparison of different orthographies for machine translation of under-resourced dravidian languages". In: *2nd Conference on Language, Data and Knowledge (LDK 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Chalabi, Achraf and Hany Gerges (2012). "Romanized arabic transliteration". In: *Proceedings of the Second Workshop on Advances in Text Input Methods*, pp. 89–96.

Chen, Song, Dana Fore, Stephanie Strassel, Haejoong Lee, and Jonathan Wright (2017). "BOLT Egyptian Arabic sms/chat and transliteration LDC2017T07". In: *Philadelphia: Linguistic Data Consortium*.

Conneau, Alexis, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou (2017). "Word translation without parallel data". In: *arXiv preprint arXiv:1710.04087*.

Darwish, Kareem (Oct. 2014). "Arabizi Detection and Conversion to Arabic". In: *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 217–224. DOI: 10.3115/v1/W14-3629. URL: https://aclanthology.org/W14-3629.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Eskander, Ramy, Mohamed Al-Badrashiny, Nizar Habash, and Owen Rambow (Oct. 2014). "Foreign Words and the Automatic Processing of Arabic Social Media Text Written in Roman Script". In: *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Doha, Qatar: Association for Computational Linguistics, pp. 1–12. DOI: 10.3115/v1/W14-3901. URL: https://aclanthology.org/W14-3901.

Farhan, Wael, Bashar Talafha, Analle Abuammar, Ruba Jaikat, Mahmoud Al-Ayyoub, Ahmad Bisher Tarakji, and Anas Toma (2020). "Unsupervised dialectal neural machine translation". In: *Information Processing & Management* 57.3, p. 102181.

Farrag, Mona (2012). "Arabizi: a writing variety worth learning? an exploratory study of the views of foreign learners of Arabic on Arabizi." In.

Ganin, Yaroslav, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky (2016). "Domain-adversarial training of neural networks". In: *The journal of machine learning research* 17.1, pp. 2096–2030.

Garcia, Xavier, Aditya Siddhant, Orhan Firat, and Ankur P Parikh (2020). "Harnessing multilinguality in unsupervised machine translation for rare languages". In: *arXiv preprint arXiv:2009.11201*.

Gibson, Michael (2015). "A framework for measuring the presence of minority languages in cyberspace". In: *Linguistic and Cultural Diversity in Cyberspace*, p. 61.

González-Carvajal, Santiago and Eduardo C Garrido-Merchán (2020). "Comparing BERT against traditional machine learning text classification". In: *arXiv preprint arXiv:2005.13012*.

Guellil, Imane, Faiçal Azouaou, Mourad Abbas, and Sadat Fatiha (2017). "Arabizi transliteration of algerian arabic dialect into modern standard arabic". In: *Social MT*, pp. 1–8.

Habash, Nizar, Mona Diab, and Owen Rambow (May 2012). "Conventional Orthography for Dialectal Arabic". In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), pp. 711–718. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/579_Paper.pdf.

Haeri, Niloofar (2003). *Sacred language, ordinary people: Dilemmas of culture and politics in Egypt*. Springer.

Horesh, Uri and William M Cotter (2016). "Current research on linguistic variation in the Arabic-speaking world". In: *Language and Linguistics Compass* 10.8, pp. 370–381.

Al-Jallad, Ahmad (2014). "On the genetic background of the Rbbl bn Hfm grave inscription at Qaryat al-Fāw1". In: *Bulletin of the School of Oriental and African Studies* 77.3, pp. 445–465.

Jaran, Samia A and Fawwaz Al-Abed Al-Haq (2015). "The Use of Hybrid Terms and Expressions in Colloquial Arabic among Jordanian College Students: A Sociolinguistic Study." In: *English Language Teaching* 8.12, pp. 86–97.

Jeblee, Serena, Weston Feely, Houda Bouamor, Alon Lavie, Nizar Habash, and Kemal Oflazer (2014). "Domain and dialect adaptation for machine translation into egyptian arabic". In: *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pp. 196–206.

Junczys-Dowmunt, Marcin, Tomasz Dwojak, and Hieu Hoang (2016). "Is neural machine translation ready for deployment? A case study on 30 translation directions". In: *arXiv preprint arXiv:1610.01108*.

Keong, Yuen Chee, Othman Rahsid Hameed, and Imad Amer Abdulbaqi (2015). "The use of Arabizi in English texting by Arab postgraduate students at UKM". In.

Al-Khatib, Mahmoud and Enaq H Sabbah (2008). "Language choice in mobile text messages among Jordanian university students". In: *SKY Journal of Linguistics* 21.1, pp. 37–65.

Koneru, Sai, Danni Liu, and Jan Niehues (Apr. 2021). "Unsupervised Machine Translation On Dravidian Languages". In: *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Kyiv: Association for Computational Linguistics, pp. 55–64. URL: https://aclanthology.org/2021.dravidianlangtech-1.7.

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

Lample, Guillaume, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato (2018). *Unsupervised Machine Translation Using Monolingual Corpora Only*. arXiv: 1711.00043 [cs.CL].

Lample, Guillaume, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato (2018). "Phrase-based & neural unsupervised machine translation". In: *arXiv preprint arXiv:1804.07755*.

Li, Zuchao, Hai Zhao, Rui Wang, Masao Utiyama, and Eiichiro Sumita (Nov. 2020). "Reference Language based Unsupervised Neural Machine Translation". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Compu-

tational Linguistics, pp. 4151–4162. DOI: 10.18653/v1/2020.findings-emnlp.371. URL: https://aclanthology.org/2020.findings-emnlp.371.

Lison, P. and J. Tiedemann (2016). *OpenSubtitles: Extracting Large Parallel Corpora from Movie and TV Subtitles*.

Liu, Yinhan, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer (2020). "Multilingual denoising pre-training for neural machine translation". In: *Transactions of the Association for Computational Linguistics* 8, pp. 726–742.

Lulu, Leena and Ashraf Elnagar (2018). "Automatic Arabic dialect classification using deep learning models". In: *Procedia computer science* 142, pp. 262–269.

Malmasi, Shervin, Eshrag Refaee, and Mark Dras (2015). "Arabic dialect identification using a parallel multidialectal corpus". In: *International conference of the pacific association for computational linguistics*. Springer, pp. 35–53.

Masmoudi, Abir, Mariem Ellouze Khmekhem, Mourad Khrouf, and Lamia Hadrich Belguith (2019). "Transliteration of Arabizi into Arabic script for Tunisian dialect". In: *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 19.2, pp. 1–21.

May, Jonathan, Yassine Benjira, and Abdessamad Echihabi (2014). "An Arabizi-English social media statistical machine translation system". In: *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas: MT Researchers Track*, pp. 329–341.

Moosa, Ibraheem Muhammad, Mahmud Elahi Akhter, and Ashfia Binte Habib (May 2023). "Does Transliteration Help Multilingual Language Modeling?" In: *Findings of the Association for Computational Linguistics: EACL 2023*. Dubrovnik, Croatia: Association for Computational Linguistics, pp. 670–685. URL: https://aclanthology.org/2023.findings-eacl.50.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318.

Raytheon, BBN (2012). "Technologies, Linguistic Data Consortium, and Sakhr Software. 2012". In: *Arabic-Dialect/English Parallel Text (LDC2012T09). Linguistic Data Consortium*.

Saâdane, Houda, Hosni Seffih, Christian Fluhr, Khalid Choukri, and Nasredine Semmar (May 2018). "Automatic Identification of Maghreb Dialects Using a Dictionary-Based Approach". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1575.

Sajjad, Hassan, Alexander Fraser, and Helmut Schmid (2012). "A statistical model for unsupervised and semi-supervised transliteration mining". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 469–477.

Salloum, Wael, Heba Elfardy, Linda Alamir-Salloum, Nizar Habash, and Mona Diab (2014). "Sentence level dialect identification for machine translation system selection". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 772–778.

Samih, Younes, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio (2016). “Multilingual code-switching identification via lstm recurrent neural networks”. In: *Proceedings of the second workshop on computational approaches to code switching*, pp. 50–59.

Shazal, Ali, Aiza Usman, and Nizar Habash (Dec. 2020). “A Unified Model for Arabizi Detection and Transliteration using Sequence-to-Sequence Models”. In: *Proceedings of the Fifth Arabic Natural Language Processing Workshop*. Barcelona, Spain (Online): Association for Computational Linguistics, pp. 167–177. URL: https://aclanthology.org/2020.wanlp-1.15.

Sun, Chi, Xipeng Qiu, Yige Xu, and Xuanjing Huang (2019). “How to fine-tune bert for text classification?” In: *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*. Springer, pp. 194–206.

Talafha, Bashar, Mohammad Ali, Muhy Eddin Za’ter, Haitham Seelawi, Ibraheem Tuffaha, Mostafa Samir, Wael Farhan, and Hussein T Al-Natsheh (2020). “Multi-dialect arabic bert for country-level dialect identification”. In: *arXiv preprint arXiv:2007.05612*.

Tobaili, Taha (Aug. 2016). “Arabizi Identification in Twitter Data”. In: *Proceedings of the ACL 2016 Student Research Workshop*. Berlin, Germany: Association for Computational Linguistics, pp. 51–57. DOI: 10.18653/v1/P16-3008. URL: https://aclanthology.org/P16-3008.

Tracey, Jennifer et al. (2021). “BOLT Egyptian Arabic SMS/Chat Parallel Training Data LDC2021T15”. In: *Web Download. Linguistic Data Consortium, Philadelphia*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). “Attention is all you need”. In: *Advances in neural information processing systems* 30.

Virtanen, Antti, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo (2019). “Multilingual is not enough: BERT for Finnish”. In: *arXiv preprint arXiv:1912.07076*.

Wees, Marlies van der, Arianna Bisazza, and Christof Monz (Dec. 2016). “A Simple but Effective Approach to Improve Arabizi-to-English Statistical Machine Translation”. In: *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 43–50. URL: https://aclanthology.org/W16-3908.

Yaghan, Mohammad Ali (2008). “" Arabizi": A Contemporary Style of Arabic Slang”. In: *Design Issues* 24.2, pp. 39–52.

Younes, Jihen, Hadhemi Achour, and Emna Souissi (2015). “Constructing linguistic resources for the Tunisian dialect using textual user-generated contents on the social web”. In: *Current Trends in Web Engineering: 15th International Conference, ICWE 2015 Workshops, NLPIT, PEWET, SoWEMine, Rotterdam, The Netherlands, June 23-26, 2015. Revised Selected Papers 15*. Springer, pp. 3–14.

Zaidan, Omar F. and Chris Callison-Burch (Mar. 2014). “Arabic Dialect Identification”. In: *Computational Linguistics* 40.1, pp. 171–202. DOI: 10.1162/COLI_a_00169. URL: https://aclanthology.org/J14-1006.

Zakraoui, Jezia, Moutaz Saleh, Somaya Al-Maadeed, and Jihad Mohamed Alja’am (2021). “Arabic Machine Translation: A Survey With Challenges and Future Directions”. In: *IEEE Access* 9, pp. 161445–161468. DOI: 10.1109/ACCESS.2021.3132488.

Zampieri, Marcos, Liling Tan, Nikola Ljubešic´, and Jörg Tiedemann (Aug. 2014). "A Report on the DSL Shared Task 2014". In: *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University, pp. 58–67. DOI: 10.3115/v1/W14-5307. URL: https://aclanthology.org/W14-5307.

Zbib, Rabih, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar Zaidan, and Chris Callison-Burch (2012). "Machine translation of Arabic dialects". In: *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pp. 49–59.

Zhu, Jinhua, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu (2020). "Incorporating bert into neural machine translation". In: *arXiv preprint arXiv:2002.06823*.

8 Appendix

8.1 Code

Listing 8.1: tokenization process using BertTokenizer

```
tokenizer = BertTokenizer.from_pretrained("aubmindlab/bert-base-arabertv02-twitter")
    #"bert-base-multilingual-cased" in case of Multilingual BERT
input_ids = []
attention_masks = []

for sent in sentences:
  encoded_dict = tokenizer.encode_plus(
                        sent,                      # Sentence to encode.
                        truncation=True,
                        add_special_tokens = True, # Add '[CLS]' and '[SEP]'
                        max_length = 64,           # Pad and truncate all sentences.
                        padding = 'max_length',
                        return_attention_mask = True,   # Construct attention masks.
                        return_tensors = 'pt',          # Return pytorch tensors.
                        )
  # Add the encoded sentence to the list.
  input_ids.append(encoded_dict['input_ids'])

  # Add its attention mask
  attention_masks.append(encoded_dict['attention_mask'])
```

Listing 8.2: Implementation of the model's fine-tuning loop

```
for epoch_i in range(0, epochs):
        total_train_loss = 0
        model.train()
        # 'batch' contains three pytorch tensors:
        #   [0]: input ids
        #   [1]: attention masks
        #   [2]: labels
        for step, batch in enumerate(train_dataloader):
                b_input_ids = batch[0].to(device)
                b_input_mask = batch[1].to(device)
                b_labels = batch[2].to(device)
                model.zero_grad()
                result = model(b_input_ids,
                token_type_ids=None,
                attention_mask=b_input_mask,
                labels=b_labels,
                return_dict=True)
                loss = result.loss
                logits = result.logits
                total_train_loss += loss.item()
                loss.backward()
                torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)
                optimizer.step()
                scheduler.step()
        avg_train_loss = total_train_loss / len(train_dataloader)
```