نموذج التفويض

أنا الاء محمد مصلح السواعير، أفوض الجامعة الأردنية بتزويد نسخ من رسالتي /أطروحتي للمكتبات، أو المؤسسات، أو الهيئات، أو الأشخاص عند طلبهم حسب التعليمات النافذة ف الجامعة.

التوقيع:

التاريخ: 2024/6/12

The University of Jordan

Authorization Form

I, Ala' Mohammed Musleh Alsaweir, authorize the University of Jordan to supply copies of my Thesis/ Dissertation to libraries or establishments or individuals on request, according to the University of Jordan regulations.

Signature:

Date: 12/6/2024

RECOGNIZING TEXTS OF JORDANIAN ARABIC DIALECT USING MACHINE LEARNING TECHNIQUES

By
Ala' Mohammed Alsaweir

Supervisor
Dr. Gheith Abandah, Prof.

This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master's Degree of Science in Computer and Network Engineering

School of Graduate Studies
The University of Jordan

June, 2024

COMMITTEE DECISION

This Thesis/Dissertation (Recognizing Texts of Jordanian Arabic Dialect Using Machine Learning Techniques) was Successfully Defended and Approved on

Examination Committee	**Signature**
Dr. Gheith Abandah (Supervisor) Prof. of Computer Engineering	----------
Dr. Ramzi Saifan, (Member) Associate Prof. of Computer Engineering	----------
Dr. Mousa Al-Akhras, (Member) Associate Prof. of Artificial Intelligence	----------
Dr. Nailah Al-Madi, (Member) Associate Prof. of Data Science	----------

DEDICATION

A beautiful feeling filled with mixed emotions dominated by joy and the tear of reaching this moment, the moment of writing this dedication.

I dedicate it to my strong and ambitious self, to my mother Fatima who has always enveloped me in her prayers, to my support, my father Mohammed.

To my husband, Mustafa, thank you for your constant presence by my side and your support.

To my children and my beautiful world, Nadia, Qais, and Salma, who illuminate my life with their laughter and sincere emotions, giving me hope and perseverance.

I dedicate it to my brothers, sisters, and friends, to everyone who has loved me and wished me well.

AKNOWLEDGEMENT

Extend my deepest gratitude to my advisor, Professor Gheith Abandah, whose invaluable assistance has been paramount in the completion of this thesis. Your guidance, expertise, and steadfast support have played a pivotal role in my academic journey. Your mentorship has not only enriched my understanding of the subject matter but has also inspired me to strive for excellence in all endeavors. Your constructive feedback has been instrumental in refining my work, and your patience and encouragement have motivated me to exceed my own expectations.

I am truly grateful for the time and dedication you have generously devoted to helping me succeed. Your unwavering belief in my abilities has empowered me to overcome challenges and achieve my goals.

Table of Content

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADI	Arabic Dialect Identification
ANNs	Artificial Neural Networks
AOC	Arabic Online Commentary Dataset
BERT	Bidirectional Encoder Representations from Transformers
BGRU	Bidirectional Gated Recurrent Unit
BiLSTM	Bidirectional Long Short-Term Memory
BOW	Bag-of-Words
BTEC	Basic Traveling Expression Corpus
C4	Colossal Clean Crawled Corpus
CA	Classical Arabic
CGRU	CNN Gated Recurrent Unit
CLSTM	Convolutional LSTM
CNN	Convolutional Neural Networks
DA	Dialectal Arabic
DL	Deep Learning
DNN	Deep Neural Networks
DT	Decision Trees
EG	Egyptian Dialect
EGY	Egyptian
GCS	Google Cloud Storage
GLF	Gulf
GloVe	Global Vectors for Word Representation
GPUs	Graphic Processing Units

GRU	Gated Recurrent Unit
HANGRU	Hierarchical Attention Network Gated Recurrent Unit
HANLSTM	Hierarchical Attention Network Long Short-Term Memory
JO	Jordanian Dialect
JODA	Jordanian Dataset
KNN	K-Nearest Neighbor
LEV	Levantine
LSTM	Long Short-Term Memory
MADAR	Multi Arabic Dialect Applications and Resources
MNB	Multinomial Naive Bayes
MSA	Modern Standard Arabic
mT5	Multilingual T5
MTL	Multi-Task Learning
NADI	Nuanced Arabic Dialect Identification Shared Task
NB	Naïve Bayes
NLP	Natural Language Processing
OOV	Out-of-Vocabulary
RAM	Random Access Memory
RF	Random Forest
RNNs	Recurrent Neural Networks
SA	Saudi Arabia Dialect
SDC	Shami Dialect Corpus
SVM	Support Vector Machine
SY	Syrian Dialect
TF-IDF	Frequency-Inverse Document Frequency

Recognizing Texts of Jordanian Arabic Dialect Using Machine Learning Techniques

By

Ala' Mohammed Alsaweir

Supervisor

Dr. Gheith Abandah, Prof.

ABSTRACT

Due to the recent social media revolution, spoken Arabic language, often known as dialectal Arabic, has begun to appear in written form. Dialect refers to a set of linguistic structures and vocabulary specific to a particular environment. Dialects also vary in the pronunciation of certain words, which distinguishes a group of individuals from others. Among the authentic Arabic dialects, the Jordanian dialect stands out, considered one of the Levantine dialects, which also branches into various subdivisions depending on geographical regions. Recognizing Arabic dialects stands as the primary preprocessing element for any natural language processing (NLP) task, including social media analysis and machine translation. Dialect identification might be considered more challenging from the language identification problem itself since the dialects could be thought of as closely related languages, so the automatic determination of an Arabic text's dialect remains a significant challenge for researchers. Transformers hold significant influence within today's deep learning (DL) framework. Transformers are widely used and have had a significant impact in many fields, playing a significant role in NLP tasks, including language identification. The pre-trained Byte-to-Byte model (ByT5) is based on the T5 architecture, which is a standard Transformer architecture. ByT5 is a token-free model

designed to process text directly as raw bytes, eliminating the need for tokenization, operates on UTF-8 encoded characters, offering benefits such as language agnosticism, robustness to noise, and reduced preprocessing complexity. The various Arabic dialect texts were gathered from multiple dataset sources: Arabic Online Commentary (AOC), Shami Dialect Corpus (SDC), and the Multi-Arabic Dialect Applications and Resources (MADAR) project, containing about 20k Jordanian dialect texts. We conducted several experiments using a deep learning ByT5 model to address the challenge of recognizing the Jordanian dialect among various Arabic dialects. The model achieved very good accuracy in identifying the Jordanian dialect, boasting an F1 score of 88.7% from the multi-class classification experiment. Additionally, we improved this result by using the Jordanian Dataset (JODA), which has about 39k Jordanian dialect sequences. Doubling the Jordanian dialect dataset size from 20k to 39k sequences improved the Jordanian dialect detection F1 score from 88.7% to 95.3%.

Chapter One

Introduction

This chapter provides an overview of the background of the thesis, the research problem, the importance of the research, the objectives of the thesis, the questions guiding the thesis, the contributions of the thesis, the methodology employed in the thesis, and an outline of the thesis.

1.1 Background

Arabic is one of the Semitic languages spoken by approximately 313 million individuals as their primary language and by an additional 270 million speakers as a second language. It serves as the first language for 22 Arab countries (Algihab *et al.*, 2019). The Arabic language encompasses three main types: classical, modern, and dialectal. Classical Arabic (CA), the language of the holy book of Islam (the Quran), holds significant religious and historical importance. Modern Standard Arabic (MSA), derived from classical Arabic, has evolved by dropping certain linguistic features, such as diacritics. Also referred to as Al-Arabiyya, Al-Fusha, and Literary Arabic, MSA finds extensive usage in modern literature, education, and news broadcasting. Additionally, modern Arabic exhibits various dialects, each tailored for specific social contexts, resulting in lexical, morphological, and grammatical variations across regions (Algihab *et al.*, 2019).

The spoken Arabic language, often known as dialectal Arabic, differs throughout the Arab world. These dialects started to appear in written online social interactions because of social media's explosive growth. Arabic dialects differ from one another and are influenced by the speakers' social background and geographic region. They are commonly divided into five major groups: the Levantine (Palestine, Syria, Lebanon, Jordan), Iraqi (Iraq), Maghreb (Libya, Tunisia, Algeria, Morocco, and western Sahara),

Egyptian (Egypt and some parts of Sudan), and Gulf (Qatar, Kuwait, Bahrain, the United Arab Emirates, Saudi Arabia, Oman, and Yemen) (Abu Kwaik *et al*., 2018). Table 1 illustrates a single sentence expressed in various Arabic dialects as well as MSA. This sentence is drawn from the Multi-Arabic Dialect Applications and Resources (MADAR) project, which has curated a substantial parallel corpus comprising 25 Arabic city dialects within the travel domain (Bouamor *et al.*, 2018). This corpus was constructed by translating specific sentences from the Basic Traveling Expression Corpus (BTEC), available in both French and English, into diverse dialects. BTEC serves as a multilingual spoken language corpus containing tourism-oriented phrases commonly found in travel phrasebooks intended for tourists traveling abroad.

Table 1. Sample of Same Text in Different Dialects.

Type of Text	Text
Jordanian dialect	هيو هناك، بالزبط مقابل مكتب المعلومات السياحية
Egyptian dialect	ده قدامك هناك، يادوبك قدام مكتب استعلامات السياحة
Syrian dialect	موجود هنيك، قدام مكتب معلومات السياح بالزبط
Saudi Arabia dialect	هناك، بالضبط مقابل مكتب معلومات السياح
MSA	هناك ، أمام بيانات السائح تماما

Table 1 reveals that all Arabic dialects utilize the same character set, and a significant portion of their vocabulary is shared across different varieties. Consequently, distinguishing and segregating the dialects from one another is not a straightforward task.

Jordanian Arabic is the colloquial form of Arabic spoken in Jordan, falling within the Levantine dialect. It is categorized as a low-resource language, lacking widely available linguistic tools such as written texts, pronunciation dictionaries, and morphological analyzers. Moreover, Jordanian Arabic is recognized for its morphological complexity, characterized by extensive inflection and derivation, resulting in a myriad of surface forms stemming from a single root. Additionally, regional

variations exist within Jordanian Arabic, with distinctions between urban, rural, and Bedouin dialects observable across different cities in Jordan.

Abandah *et al.* (2015) delved into the usage of the Arabic language within Jordanian social networking and mobile phone communications, aiming to understand its role and perception in digital platforms. Through an examination of linguistic features, communication patterns, and user attitudes, they sought insights into Arabic's impact in the digital sphere. The study found colloquial Jordanian Arabic to be the most prevalent (55.4%), followed by standard Arabic (36.4%), with a minority employing standard Arabic alongside colloquial words (8.2%). Notably, Messages and Facebook emerged as platforms with the highest prevalence of colloquial and mixed dialects (79.3% and 75.3%), indicative of their informal communication nature. Conversely, News and Blogs predominantly utilized standard Arabic (67.2% and 56.9%). However, within Blogs, excluding comments, the usage skewed slightly towards standard Arabic at 62%, with 27% mixed and 11% colloquial. The elevated presence of mixed language in news suggests a preference for standard Arabic, occasionally interspersed with colloquial words for clearer expression of ideas and emotions.

Text classification is the process of identifying the categories of text data by using features that are extracted from the raw text data. Text data differs from numerical, image, or signal data, requiring careful processing using Natural Language Processing (NLP) techniques. The task of language identification is essentially a document classification task, wherein documents are classified into classes or categories denoted by a finite set of labels.

Over the last decade, there has been a remarkable increase in interest in dialectal Arabic (DA), resulting in an excitement for NLP research. The primary reason for this is that political factors enabled researchers to receive funding. Technically, data was

abundant due to the widespread usage of Arabic dialects on social media. This, together with developments in machine learning, attracted academics to it all (Elnagar *et al.*, 2021).

Arabic dialect identification (ADI) is a specific task of NLP, aiming to automatically predict the Arabic dialect of a given text. ADI is the first step in various NLP applications such as error correction, machine translation, multilingual text-to-speech synthesis, and cross-language text generation. ADI in written texts is the process of building a recognizer that is able, given an Arabic piece of text (e.g., sentence, paragraph, document), to determine if the text was written in dialectal Arabic and which dialect was written (Althobaiti, 2020).

Deep learning (DL), a subset of machine learning, uses multiple layers to extract higher-level features from raw data, achieving impressive results in NLP tasks like question classification and sentiment analysis. Transformer-based models, now state-of-the-art in text recognition, excel due to their parallel processing capability, efficient handling of long-range dependencies, and lower computational costs. These factors have propelled transformer architecture to the top of various NLP leaderboards (StankeviǸcius *et al.*, 2022). We utilized the pre-trained Byte-to-Byte model (ByT5) to recognize the Jordanian dialect.

1.2 Research Problem

The challenges encountered in identifying Arabic dialect texts arise from the close relationship between Arabic dialects, shared vocabulary, and notable linguistic disparities between MSA and various dialects. Identifying Arabic dialects poses several challenges rooted in their shared linguistic characteristics and nuanced differences from MSA. These difficulties include the similarity in linguistic traits among dialects and the use of the same set of letters, which complicates their differentiation based on traditional linguistic

features (Althobaiti, 2020). Moreover, pronunciation variations exist across dialects, with some incorporating sounds that are not present in the Arabic alphabet. This leads to divergent representations of certain sounds, sometimes necessitating the adoption of new letters borrowed from other alphabets. Consequently, the complexity and variability between dialects increase (Harrat *et al.,* 2019; Althobaiti, 2020). The aim of this thesis is to develop a solution capable of handling the complexities of Arabic dialect identification especially Jordanian dialect and providing viable solutions for this essential task.

1.3 Research Importance

Automatic dialect classification is important for several reasons. Firstly, automatic dialect identification can be used to find challenging cases for dialectology research or for the evaluation of downstream NLP systems. Through the process of rating documents according to the density of dialect, researchers can identify instances in which dialect features are particularly dense or rare. This may help in enhancing the efficiency of NLP systems that come after or in better understanding the linguistic characteristics of various dialects or languages. (Demszky *et al.*, 2020).

Secondly, language identification is also an enabling technology that can help automatically filter foreign text in some tasks, acquire multilingual data, and enhance tasks like machine translation.

Modern Arabic-to-English machine translation systems translate MSA sentences quite effectively, but when the input is dialect, they frequently yield confusing results. For example, in Figure 1, two similar Arabic sentences in essence, one in MSA and one in Levantine Arabic, are translated by the same MT system into English. The figure shows an acceptable translation from MSA rather than the dialect sentence (Zaidan and Callison-Burch,2011).

Figure 1. Challenges in Dialectal Arabic-to-English Machine Translation (Zaidan and Callison-Burch, 2011).

Also, recognizing Jordanian Arabic dialect using machine learning techniques is important to correct the errors and translate it to modern standard Arabic, which can help non-Jordanian people, beginner readers, and kids with dyslexia understand the Jordanian Arabic dialect.

1.4 Thesis Objectives

The primary objective of this thesis is to leverage a DL model capable of delivering dependable and precise predictions to identify the Jordanian dialect among multiple dialects.

1.5 Thesis Questions

This study targets answering the following questions:

- Which machine learning algorithm is suitable for Arabic dialect identification (ADI) to recognize Jordanian Arabic?
- How can we address the current lack of training data for the Jordanian dialect?
- What are the future directions and potential areas of research in the field of Jordanian dialect identification using machine learning, and how might these developments benefit the larger field of Arabic language processing and analysis?

- How does data augmentation help deep learning models for Jordanian dialect detection become more robust and broadly applicable?

1.6 Thesis Contributions

This thesis used the ByT5 DL model, as detailed in Chapter 3, aimed at classifying the Jordanian dialect. Notably, this research represents the pioneering effort to classify the Jordanian dialect using an acceptable amount of data collected from multiple sources, a unique aspect highlighted within this thesis. Until the writing of this thesis, prior studies have not endeavored to utilize DL models specifically tailored for predicting the Jordanian dialect. Therefore, this work stands as the inaugural attempt to develop a DL model explicitly designed for classifying this dialect. The proposed model demonstrates efficient and accurate predictions while requiring minimal computational resources, thereby offering a cost-effective solution.

1.7 Thesis Methodology

The outlined steps delineate our strategy to fulfill the objectives of this thesis:

- Gathering and obtaining the Arabic dialect dataset, especially the Jordanian dialect, from several sources.
- Conducting a comprehensive review of prior studies and relevant literature pertaining to our thesis topic, with emphasis on recent research, summarizing the methodologies employed in these studies.
- Employing a range of techniques and tools to assess the dataset's suitability for DL applications. This process involves identifying and rectifying issues such as noise and missing values to ensure the dataset's integrity and reliability for utilization in DL models.

- Preparing the dataset by segmenting it appropriately for training purposes, including partitioning it for experimentation, and creating a unified test set to evaluate all experiments using the same dataset.
- Utilizing the ByT5 model to identify the Jordanian dialect effectively.
- Training the proposed model on the entire dataset and evaluating its performance using accuracy and F1 score metrics.
- Reporting and documenting our work and presenting the results.

1.8 Thesis Outline

This thesis includes related works for the research in Chapter 2. Chapter 3 includes a theoretical background for related models, neural networks, definitions, and tools that I used in my experiments. Chapter 4 offers a detailed description of the dataset utilized and outlines the methodology employed in the thesis. Chapter 5 encompasses the presentation, evaluation, and discussion of training results, along with an overview of the experimental setup and environment. Finally, Chapter 6 concludes the thesis and outlines avenues for future research.

Chapter Two

Literature Review

2.1 Introduction

This chapter explores related studies that address text classification and identification, encompassing both traditional machine learning and deep learning approaches. Additionally, an overview of some works mentioned in it pertaining to identifying the Jordanian dialect will be provided.

2.2 Text Classification Methods

Text classification is the process of identifying the categories of text data by using features that are extracted from the raw text data. Text data differs from numerical, image, or signal data, requiring careful processing using NLP techniques. The preparation of text data for modeling represents a critical initial step. Traditional models often necessitate the artificial extraction of high-quality sample features, followed by classification using classical machine learning algorithms. However, this approach tends to limit efficiency due to the constraints of feature extraction. In contrast, deep learning employs a series of nonlinear transformations, enabling the direct translation of features into outputs. This integration of feature engineering into the model fitting process distinguishes deep learning from traditional methods. A flowchart of the procedures involved in text classification is shown in the figure below, which considers both traditional and deep methods (Li *et al.,* 2022).

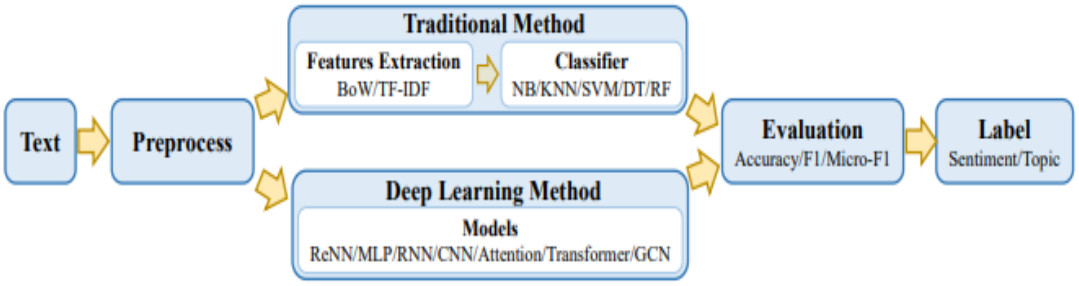

Figure 2. Traditional and Deep Learning Text Classification Techniques *(*Li *et al.*, 2022).

There have been several works reviewing text classification and identification; some of them will be mentioned under sections and subsections below.

2.2.1 Traditional Methods

Traditional methods refer to models that rely on statistics, including support vector machines (SVM), k-nearest neighbors (KNN), and Naïve Bayes (NB). When it comes to traditional models, the text classification task is first run through NB. Subsequently, general classification models—also known as classifiers—like KNN, SVM, and random forest (RF) are suggested and are frequently utilized for text classification.

Traditional models enhance text classification speed and accuracy while extending their applicability. The first step of training traditional models involves preprocessing the raw input text, which typically includes statistical analysis, data cleaning, and word segmentation. Subsequently, text representation employs methodologies such as bag-of-words (BOW), n-gram, term frequency-inverse document frequency (TF-IDF), word2vec, and global vectors for word representation (GloVe). These techniques transform preprocessed text into formats that are more computationally manageable and minimize information loss (Li *et al.*, 2022).

Luo (2021) explored the most effective machine learning techniques for accurately classifying English text documents. He conducted experiments employing a range of machine learning algorithms and features to evaluate their impact on classification

performance. These techniques encompassed SVM, NB, KNN, and decision trees (DT). Various features were examined, including term TF-IDF, word embeddings, and n-grams, to gauge their effectiveness in enhancing classification accuracy. Results indicated that SVM consistently outperformed the other machine learning techniques across the datasets utilized.

2.2.2 Deep Learning Methods

Deep Neural Networks (DNNs) are artificial neural networks crafted to emulate the intricate workings of the human brain, enabling them to autonomously discern complex features from data. This approach surpasses traditional models in tasks such as speech recognition, image processing, and text comprehension. Input datasets undergo analysis to categorize the data into single-label, multi-label, unsupervised, or imbalanced datasets. Depending on the dataset's characteristics, input word vectors are inputted into the DNN for training until a specified termination condition is met. The efficacy of the trained model is evaluated through downstream tasks such as sentiment analysis, question answering, and event prediction. In recent decades, numerous deep learning models have emerged for text classification, encompassing tasks such as sentiment analysis, topic labeling, news classification, question answering, dialog act classification, natural language inference, and relation classification (Li *et al.*, 2021).

Elnagar *et al.* (2020) constructed two new corpora tailored for Arabic classification tasks: the single-label Arabic News Articles Dataset (SANAD) and the multi-label News Articles Dataset in Arabic (NADiA). Each corpus comprises multiple datasets. They employed nine robust deep neural network-based classifiers, which were tested on extensive datasets, demonstrating high accuracy for both single- and multi-label Arabic text categorization tasks.

The deep neural network models utilized include Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), Bidirectional Gated Recurrent Units (BIGRU), Bidirectional Long Short-Term Memory (BiLSTM), CNN Gated Recurrent Units (CGRU), CNN Gated Recurrent Units (CLSTM), Hierarchical Attention Network Gated Recurrent Units (HANGRU), and Hierarchical Attention Network Long Short-Term Memory (HANLSTM).

Among these models, the top-performing ones for the single-label categorization task were CNN, BiLSTM, BiGRU, and HANGRU. These models were also trained for multi-label tasks. Notably, all models exhibited strong performance on the SANAD corpus, with HANGRU achieving the highest accuracy at 95.81%. For the NADiA corpus, CNN attained the highest accuracy, surpassing 70.34% confidence for a maximum subset of 8 categories from the SkyNewsArabia dataset. On the Masrawy dataset with a maximum subset of 10 categories, HANGRU demonstrated the highest accuracy, reaching 88.68%.

2.3 Dialect Classification and Identification Techniques

In this section, the focus will be on some of the research efforts dedicated to dialect classification and identification.

Bouamor *et al.* (2019) presented the outcomes of the MADAR shared task on Arabic fine-grained dialect identification. This shared task comprised two subtasks:

The MADAR Travel Domain Dialect Identification subtask (Subtask 1): The objective was to categorize written Arabic sentences into one of 26 labels, each representing the specific city dialect of the sentences, or MSA. The top performance was achieved by the team ArbDialectID. They achieved a macro F1-score of 67.32% (Abu Kwaik and Saad, 2019).

The MADAR Twitter User Dialect Identification subtask (Subtask 2): This subtask aimed to classify Twitter user profiles into one of 21 labels, representing 21 Arab countries, using only the tweets from the Twitter users. This shared task marked the first attempt to target a broad range of dialect labels at both the city and country levels. The data for the shared task was generated or collected within the framework of the MADAR project. A total of 21 teams from 15 countries participated in the shared task. The highest performance was achieved by the UBC-NLP team, a team led by Zhang and Abdul-Mageed (2019), which achieved a macro F1-score of 71.70%.

Abdul-Mageed *et al.* (2020) founded the first Nuanced Arabic Dialect Identification Shared Task (NADI) and presented its outcomes. The objective of this shared task is to facilitate the development of resources and foster endeavors aimed at identifying Arabic dialects within a diverse array of Arabic variants spanning 21 Arab countries and encompassing a total of 100 provinces across these nations. The shared task consists of two subtasks: country-level dialect identification (Subtask 1) and province-level sub-dialect identification (Subtask 2). To support participants, Abdul-Mageed *et al.* (2020) provided a novel Twitter-labeled dataset curated exclusively for this shared endeavor. Subtask 1 focuses on recognizing country-level dialects from concise written sentences (tweets). Participants were furnished with labeled data segregated into distinct train and development splits. Among the participants, the Mawdoo3-AI team achieved the most notable performance, attaining a 26.78% macro F1 score (Talafha *et al.*, 2020).

Talafha *et al.* (2020) presented the top-performing approach that achieved the number-one place on NADI Task 1, which focuses on identifying dialects at the country level. Their approach hinges on leveraging a Bidirectional Encoder Representations from Transformers (BERT) architecture. They capitalized on the novel dataset provided by the NADI shared task, comprising approximately 31,000 labeled tweets spanning all 21 Arab

countries, alongside an unlabeled corpus of 10 million tweets. The labeled dataset was divided into 21,000 examples for training, with the remaining 10,000 tweets equally distributed between the development and test sets. Utilizing the unlabeled dataset for further pre-training, which was sourced as crawl text from Twitter alongside the IDs of 10 million tweets (with a retrieval success rate of 97.7%), they proceeded with a three-stage approach. In the initial stage, they pre-trained a publicly available BERT model (Arabic-BERT) using the 10 million tweets supplied by the NADI organizers. Subsequently, in the second stage, they trained the resulting model on the labeled NADI data for Task 1, iterating multiple times with varying combinations of maximum sentence length and learning rate. In the final stage, they selected the four best-performing iterations, determined by their performance on the development dataset, and aggregated their softmax predictions through a straightforward element-wise averaging function to yield the final prediction for each tweet.

Abdul-Mageed *et al.* (2021) unveiled the findings from the second NADI 2021 shared task. Mirroring NADI 2020, this shared task centers on the same 21 Arab countries and 100 corresponding provinces, with a primary focus on Twitter data. However, NADI 2021 introduced four distinct subtasks: country-level MSA identification (Subtask 1.1), country-level dialect identification (Subtask 1.2), province-level MSA identification (Subtask 2.1), and province-level sub-dialect identification (Subtask 2.2).

For this collaborative endeavor, participants were provided with a new Twitter-labeled dataset, which is openly accessible for research purposes. The dataset was partitioned into MSA and Dialectal Arabic (DA) segments to enhance signal strength. The macro F1 score was adopted as the official metric across all four subtasks. Team Cairo Squad clinched the highest F1 score, reaching 32.26% in subtask 1.2 (country-level DA identification) (AlKhamiss *et al.*, 2021). Notably, many leading teams employed

transformer-based architectures. Specifically, the Cairo Squad and CS-UM6P teams, CS-UM6P team developed their systems leveraging MARBERT, a pre-trained Transformer language model tailored to Arabic dialects and social media discourse. Team Phoneme utilized AraBERT and AraELECTRA. Additionally, Team Cairo Squad incorporated adapter modules and vertical attention into their MARBERT fine-tuning process. CS-UM6P achieved 30.64% in subtask 1.2 (country-level DA identification); they employed multitask learning, fine-tuning MARBERT jointly on country-level and province-level data (El Mekki *et al.*, 2021).

El Mekki *et al.* (2021) introduced a deep learning-based system aimed at identifying both country-level and province-level variations in MSA and DA and achieved a 30.64% F1 score in subtask 1.2 (country-level DA identification). Their approach revolves around an end-to-end deep multi-task learning (MTL) model. The MTL model consists of a shared BERT encoder, two task-specific attention layers, and two classifiers. This model is tailored for multi-task learning, enabling it to concurrently predict both country- and province-level distinctions in Arabic text. The results obtained indicate that their MTL model surpasses single-task models across various subtasks, demonstrating its efficacy in addressing the complexities of Arabic language variation identification.

Abdul-Mageed *et al.* (2022) presented the findings from the third NADI 2022 shared task, which expanded its scope to encompass both dialect identification (Subtask 1) and dialectal sentiment analysis (Subtask 2) at the country level. Subtask 1 aimed to pinpoint the specific country-level dialect within a given Arabic tweet, utilizing the training, development, and test datasets of 18 countries from NADI 2021. These datasets were compiled from tweets spanning 21 Arab countries over a 10-month period in 2019. The Abdel-Salam (2022) team (rematchka) achieved the highest performance in subtask

1, attaining a macro-F1 score of 27.06. Notably, most participating teams employed transformer-based pretrained language models, such as mBERT, ArabBERT, and MARBERT, to tackle the challenges posed by the task.

Abdel-Salam (2022) secured the top position in the NADI 2022 shared task for both country-level dialect identification and sentiment analysis in dialectical Arabic. In subtask 1, they explored various methodologies, predominantly leveraging BERT-based models. They employed two key strategies: 1) fine tuning, and 2) prompting tuning. For subtask 1, they submitted three distinct systems. The first system involved Marbert prediction with prompting, while the second system comprised a weighted ensemble of all models. The third model was selected through a hard voting process, comparing the fine-tuned version of MARBERT with the prompt-based version. Their approach yielded a notable F1-macro score of 27.06 in subtask 1.

Abdul-Mageed *et al.* (2023) presented the findings from the fourth NADI 2023 shared task. NADI aims to enhance the field of Arabic NLP by providing a platform for research teams to compete in a collaborative manner under standardized conditions. The focus of NADI is on Arabic dialects, offering unique datasets and defining subtasks to enable meaningful comparisons between different methodologies. NADI 2023 concentrated on dialect identification (Subtask 1) and dialect-to-MSAachine translation (Subtasks 2 and 3). The top-performing team (NLPeople) achieved 87.27 F1 for subtask 1, 14.76 Bleu for subtask 2, and 21.10 Bleu for subtask 3. The results obtained by participating teams indicate that dialect identification remains a complex task, but various approaches, particularly those leveraging language models, can effectively handle this challenge. Similarly, translating Arabic dialects is notably difficult due to limited training data availability (Elkaref *et al.*, 2023).

Elkaref *et al.* (2023) achieved the top position in NADI 2023 by attaining an impressive F1 score of 87.27 for subtask 1, which focuses on Arabic dialect identification. Their achievement was facilitated by a comprehensive system that utilizes various components. Firstly, they employed linguistic models tailored to the Arabic language, enabling the utilization of pre-trained language models specific to Arabic. Additionally, they incorporated a clustering and retrieval method to provide additional contextual information to target sentences. Furthermore, their strategy involved fine-tuning and leveraging data from the 2020 and 2021 shared tasks. Finally, they employed an ensemble approach, aggregating predictions from multiple models for subtask 1. Their system utilized MARBERTv2 and ARABETv02-Twitter as language models. These models rely on a transformer encoder architecture to encode sequences of words into hidden states, which are subsequently fed into a feedforward network for label prediction.

2.4 Identifying the Jordanian Dialect

Zaidan and Callison-Burch (2011) curated an Arabic online commentary dataset (AOC) by collecting reader commentary from the online editions of three prominent Arabic newspapers: Al-Ghad from Jordan, Al-Riyadh from Saudi Arabia, and Al-Youm Al-Sabe' from Egypt. These newspapers represent regions where the prevalent dialects are Levantine (LEV), Gulf (GLF), and Egyptian (EGY), respectively.

Bouamor *et al.* (2018) presented the MADAR as a valuable resource for studying Arabic dialects, offering detailed linguistic annotations, metadata, and accessibility to facilitate research and analysis in the field of Arabic linguistics. This dataset includes samples from the Jordanian dialect.

Abu Kwaik *et al.* (2018) presented the Shami Dialect Corpus (SDC), which contains Palestinian, Jordanian, Syrian, and Lebanese data. They evaluated the SDC and compared it with two other corpora by conducting experiments using n-gram models and

naive Bayes classifiers. The most favorable outcomes were observed when classifying Jordanian and Lebanese dialects using a unigram word model, achieving an accuracy of 90%. This high accuracy stems from the limited lexical similarity between the two dialects. Conversely, the least favorable results were obtained when classifying all four dialects within the entire SDC, yielding a 52% accuracy rate. This lower accuracy can be attributed to the significant overlap between dialects and the dispersion of lexical items across multiple categories.

To the best of our knowledge and research, there has not been any specialized study focused on recognizing and identifying the Jordanian dialect. Therefore, this section discusses works on Arabic dialect identification and utilizes the AOC, MADAR, and SDC datasets.

Zaidan and Callison-Burch (2014) explored various machine learning models for dialect identification, including SVMs and NB classifiers. They discussed the advantages and limitations of each model in the context of Arabic dialect identification. They presented experimental results demonstrating the effectiveness of their approach in accurately identifying Arabic dialects. The highest accuracy they achieved using a unigram word model was 85.7% in a two-way classification task. They concluded that utilizing an n-gram word and character model proved to be the most suitable method for distinguishing among these dialects. Through various experiments, including two-way and multi-way classification tasks encompassing all dialects in their dataset, they established the efficacy of their proposed approach.

Lulu and Elnagar (2018) employed four DNN models to tackle the Arabic dialectal classification problem. Their research focused on binary classification experiments for each pair of the three dialects, as well as a multi-way classification experiment encompassing all dialects together. The four deep neural network models utilized were

LSTM, CNN, BLSTM, and Convolutional LSTM (CLSTM). These models were applied to the AOC benchmark dataset, specifically targeting the three most prevalent Arabic dialects: EGP, LEV, and Gulf (including Iraqi) (GLF). The resulting subset comprised approximately 33,000 sentences, evenly distributed among the three dialects. Notably, the researchers excluded MSA, the largest segment of the AOC dataset, from their comparison.

Among the four experiments conducted, the best result was achieved by the bidirectional LSTM model, reaching an accuracy of 83.8% for the EGP-GLF pair. Additionally, LSTM outperformed the other models in the multi-way classification experiment, achieving the highest accuracy across all dialects at 71.4%.

Elaraby and Abdul-Mageed (2018) conducted a benchmarking study on the AOC dataset, focusing on dialect identification using DL techniques. They examined the effectiveness of various traditional machine learning classifiers such as logistic regression, multinomial NB, and SVM, as well as six different popular DL models. These models included convolutional neural networks (CNN), long-short-term memory (LSTM), convolutional LSTM (CLSTM), BiLSTM, bidirectional gated recurrent units (BiGRU), and BiLSTM with attention.

The dataset was split into 80% for training, 10% for development, and 10% for testing. Three classification tasks were performed: binary classification (MSA vs. dialects) achieved an accuracy of 87.65%, three-way dialect classification (Egyptian vs. Gulf vs. Levantine) achieved 87.4% accuracy, and four-way variant classification (MSA vs. Egyptian vs. Gulf vs. Levantine) achieved 82.45% accuracy on the test set. The results highlighted the effectiveness of attention based BiLSTMs for this task.

Salameh *et al.* (2018) presented the first fine-grained dialect identification (DID) system covering the dialects of 25 cities from several countries, including cities within

the same country in the Arab World. Their model, a multinomial naive bayes (MNB) utilizing TF-IDF character and word features (without the KenLM language model), achieved an accuracy of 67.9% for sentences averaging 7 words in length. Notably, the accuracy surpassed 90% when considering sentences of up to 16 words to precisely identify the speaker's city.

Abu Kwaik and Saad (2019) constructed an ADI system comprising two subsystems. The initial subsystem focuses on classifying six dialects, followed by a more intricate 26-dialect classification system, which includes MSA and classifies dialects from 25 Arab towns. To enhance performance, they experimented with various combinations of skip-gram models and n-gram models (words and characters). Additionally, they incorporated the predicted label from the first model and calculated the ratio length of each input sentence alongside language modeling features. They used the MADAR data set to build and evaluate the models; their system achieved first place in the MADAR shared task (Subtask 1), achieving an accuracy of 67.29% and an F1 score of 67.32%.

Zhang and Abdul-Mageed (2019) introduced a semi-supervised approach using BERT, leveraging its ability to understand context and learn from unlabeled data. The study demonstrates the effectiveness of this approach in improving performance on dialect identification tasks compared to traditional supervised methods. Notably, their approach secured the top position in the MADAR shared task (Subtask 2), achieving a macro F1 score of 71.70% and an accuracy of 77.40%.

Fares *et al.* (2019) proposed several neural network-based models utilizing word and document representations to achieve superior results. Their models attained an F1 macro score of 65.66% on MADAR's small-scale parallel corpus, which includes 25

dialects alongside MSA. The data utilized in all proposed systems originated from one of the two parallel corpora provided by the MADAR project.

Initially, they built upon the work of Salameh and Bouamor (2018), who employed MNB. However, experiments with n-gram ranges of one to five for character features and one-gram for word features yielded only marginal improvement over the baseline (MNB), achieving an F1-score of 64.94% on the development set. Subsequently, Fares *et al.* transitioned to DNNs, introducing models such as LSTM + CharCNN, FastText embeddings + LSTM, and Baseline Ensemble, which is an ensemble of three models. They also introduced CharTFIDF + WordTFIDF + NN and the Baseline Ensemble, an ensemble of the MNB baseline and a deep learning model applied to baseline features. Among these approaches, the latter yielded the highest score. The following table presents a summary of literature reviews on identifying the Jordanian dialect.

Table 2. Summary of the Literature Review.

Authors	Year	Techniques Used	Dataset	Performance Metrics	Summary of Results
Abu Kwaik and Saad	2019	Skip-gram and n-gram models	MADAR	Macro F1 score	67.32%
				Accuracy	67.29%
Zhang and Abdul-Mageed	2019	Semi-supervised approach using BERT	MADAR	Macro F1 score	71.7%
				Accuracy	77.40%
Talafha, *et al.*	2020	BERT model	NADI shared task dataset	Macro F1 score	26.78%
El Mekki, *et al.*	2021	MTL model consists of a shared Bidirectional Encoder Representation Transformers (BERT) encoder, two task-specific attention layers, and two classifiers.	NADI shared task dataset	Macro F1 score	30.64%

Abdel-Salam	2022	An ensemble model between fine-tuned BERT-based models and various approaches of prompt-tuning.	NADI shared task dataset	Macro F1 score	27.06%
Elkaref *et al.*	2023	System of MARBERTv2 and arabertv02-twitter as language models, utilizing both regular and staged fine-tuning techniques.	NADI shared task dataset	Macro F1 score	87.27%
Zaidan and Callison-Burch	2014	Unigram word model	AOC	Accuracy	85.7%
Lulu and Elnagar	2018	Four deep neural network models (LSTM, BLSTM, CNN, CLSTM)	AOC	Accuracy	83.8% on binary classification using BLSTM
					71.4% on three-way classification using LSTM
Elaraby and Abdul-Mageed	2018	Six deep learning models (CNN, LSTM, CLSTM, BiLSTM, BiGRU, BiLSTM with attention)	AOC	Accuracy	87.65% on binary classification using BiLSTM with attention
					87.4% on three-way classification using BiLSTM with attention
					82.45% on four-way classification using BiLSTM with attention
Salameh, *et al.*	2018	Multinomial Naive Bayes (MNB) utilizing TF-IDF character + word features	MADAR	Accuracy	67.9% for sentences averaging 7 words in length
					90% when sentences up to 16 words

Fares, *et al.*	2019	Ensemble of the MNB baseline and a deep learning model	MADAR	Macro F1 score	65.66%

Chapter Three

Theoretical Background of the Proposed Model

3.1 Introduction

In this chapter, common deep neural networks used in language classification are explored. Beginning with a quick look at RNNs, LSTMs, and BiLSTM, then reviewing the Transformer model, self-attention mechanism, and its architecture. Additionally, brief coverage is given to T5 and mT5 transformers before introducing the proposed ByT5 model.

3.2 Overview of Deep Neural Networks

Artificial neural networks (ANNs) are fundamental tools in machine learning, designed to mimic the functioning of neural networks in biological brains. ANNs offer robust techniques for data analysis, dedicated to solving complex tasks through a learning process guided by predefined policies to evaluate their performance. Essentially, neural networks comprise interconnected nodes, or neurons, organized into layers, with multiple layers forming a network.

Neural networks characterized by a deep and extensive array of hidden layers are termed DNNs. The primary components of a network include the input, hidden, and output layers. While the input and output layers serve as interfaces for data entry and output, respectively, the hidden layers provide additional computational capacity to handle complex tasks beyond the capability of simpler networks. DNNs may encompass a hundred or more hidden layers, contributing to their computational prowess. DNNs have garnered attention for their remarkable accuracy, earning them a reputation as revolutionary tools in machine learning. There are many types of DNN; the difference between them is in the way they are connected (Elnagar *et al.,* 2020). For example, in computer vision, CNNs use convolutional layers to process images, while recurrent

neural networks (RNNs) use recurrent cells to handle sequential data, particularly in NLP (Islam *et al.,* 2023).

RNNs are widely utilized, but they come with notable limitations. Computation in these models is typically organized along the symbol positions of input and output sequences. This alignment, coupled with the production of hidden states (denoted as ht) based on the input at position t and the previous hidden state (ht−1), results in a sequential structure that hinders parallelization, particularly for longer sequences due to memory constraints limiting batching between examples (Vaswani *et al.,* 2017).

The crux of the issue lies in conventional networks' short-term dependencies, which are plagued by vanishing and exploding gradients. However, achieving effective results in NLP necessitates capturing long-term dependencies. Moreover, the sequential nature of data processing and the computational approach in recurrent neural networks contribute to slow training.

The introduction of the LSTM variant of recurrent networks aimed to address these challenges. LSTMs extend the memory range of NLP tasks and alleviate the gradient descent problem inherent in conventional recurrent neural networks. Nonetheless, the persistence of sequential processing remains a hurdle for LSTMs, impeding the extraction of true context meaning. To overcome this obstacle, bidirectional LSTMs were devised. These models analyze natural language in both left-to-right and right-to-left directions, subsequently concatenating the outcomes to deduce the genuine context meaning. However, this method still entails a minor loss of the context's actual meaning (Islam *et al.,* 2023). The next section illustrates an overview of transformers and their architecture.

3.3 Transformers

In various tasks, attention mechanisms are now an essential component of compelling sequence modeling and transduction models, enabling the modeling of

dependencies regardless of how far apart they are in the input or output sequences. However, such attention processes are employed in conjunction with a recurrent network in all but a few cases (Vaswani *et al.*, 2017).

Transformers, a variant of DNNs, provide a solution to the limitations inherent in sequence-to-sequence (seq-2-seq) architectures. These limitations include the sequential processing of input and the short-term dependence of sequence inputs that prevent parallel training of networks. With their multi-head self-attention mechanism, transformers demonstrate significant potential for applications in NLP. Unlike traditional recurrent approaches, transformers employ encoding and decoding blocks that leverage attention to learn from entire sequence segments. This attention mechanism enables Transformers to capture comprehensive contextual understanding, presenting a notable advantage over LSTM and RNNs. Transformers can also compute tasks with large inputs more quickly because, unlike recurrent networks, they can operate in parallel and can be calculated using graphics processing units (GPUs) (Islam *et al.*, 2023).

3.3.1 Transformer Architecture

The transformer model was first introduced in 2017 for machine translation work by Vaswani *et al.* (2017). Since then, many other models have been created and developed to address various problems in numerous industries, drawing inspiration from the original transformer model. Some models have solely utilized the encoder or decoder module of the transformer model, while others have fully embraced the standard transformer architecture. Consequently, the function and performance of transformer-based models can vary depending on the specific architecture used. However, self-attention is an essential and frequently utilized component of transformer models, necessary for their operation. The self-attention mechanism and multi-head attention, which typically constitute the main learning layer of the architecture, are employed by

all transformer-based models. In transformer models, the attention mechanism plays a critical role because of the importance of self-attention (Vaswani *et al.*, 2017). The following figure illustrates the overall architecture using stacked self-attention and pointwise, fully connected layers for both the encoder and decoder for the transformer.

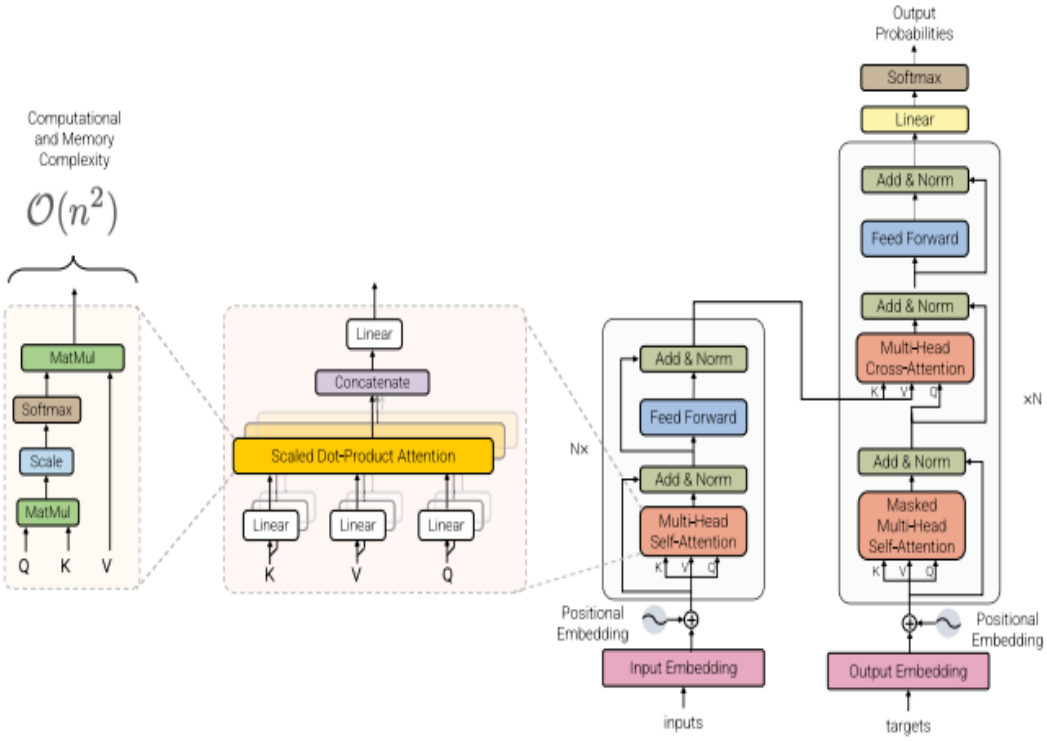

Figure 3. The Architecture of the Standard Transformer Model (Tay *et al.*, 2022).

Transformers are constructed by stacking blocks on top of one another to create multi-layered structures. Transformer blocks are characterized by residual connectors, layer normalization modules, a position-wise feed-forward network, and a multi-head self-attention mechanism. Typically, the batch size (B) and sequence length (N) are represented by a tensor of shape $R^B \times R^N$, which serves as the input for the transformer model. Initially, the input undergoes processing by an embedding layer, producing $R^B \times R^N \times R^{dmodel}$ dimensional embeddings from each one-hot token representation. Subsequently, positional encodings are added to the new tensor before passing it through a multi-headed self-attention module. Positional encodings can either be trainable

embeddings or sinusoidal inputs. The multi-headed self-attention module's input and output are connected by layer normalization layers and residual connectors.

The output of the multi-headed self-attention module is then fed into a two-layered feed-forward network. Layer normalization is employed to establish residual connections between the network's inputs and outputs. The formula for the sub-layer residual connectors with layer norm is:

$$X = LayerNorm\big(F_S(X)\big) + X \quad (1)$$

where FS is the sub-layer module, which is either the multi-headed self-attention or the position wise feed-forward layers (Tay *et al.,* 2022).

3.3.2 Utilization of Encoder and Decoder Modules

The fundamental elements of the architecture used for sequence transduction activities, such as machine translation and other NLP tasks, are the encoder and decoder in the transformer. The input sequence is processed by the encoder to extract features, which are then utilized by the decoder to generate an output sequence. Both components employ residual connections, layer normalization, and self-attention processes to support information flow and learning inside the model.

- **Encoder**: The encoder, comprising a stack of N = 6 identical layers with two sub-layers each, facilitates feature extraction from the input sequence. The first sub-layer consists of multi-head self-attention mechanisms, while the second sub-layer comprises position-wise fully connected feed-forward networks. Residual connections surround each sub-layer, followed by layer normalization. The output of each sub-layer is normalized using LayerNorm (x + Sublayer(x)). All sub-layers of the model yield output with a dimension of $d_{model} = 512$.
- **Decoder**: To generate an output sequence, the decoder utilizes the encoder's output. Like the encoder, the decoder comprises a stack of N = 6 identical layers, with three

sub-layers in each decoder layer: two sub-layers resembling those in the encoder and an additional sub-layer for multi-head attention over the output of the encoder stack. The decoder's self-attention sub-layer is modified to prevent positions from attending to subsequent positions. This adjustment, along with positional encodings, ensures that predictions for a position depend only on known outputs at positions less than that position.

3.3.3 Attention Mechanism

In the transformer model, attention serves as the central mechanism enabling the model to concentrate on pertinent segments of the input sequence during processing. This attention mechanism replaces traditional recurrent and convolutional layers, making the model more parallelizable and efficient. The attention mechanism in transformers, as proposed by Vaswani *et al.* (2017), is a fundamental concept that revolutionized neural network architectures, particularly in NLP tasks.

Mapping a query and a collection of key-value pairs to an output, where the query, keys, values, and output are all vectors, is a common representation of an attention function. The output is calculated as a weighted sum of the values, with each value's weight determined by the compatibility of the query with its corresponding key (Vaswani *et al.*, 2017). The particular attention called "Scaled Dot-Product Attention" is illustrated on the left side of Figure 3.

3.3.3.1 Multi-Head Attention

Transformers utilize multi-head self-attention, where multiple attention mechanisms operate in parallel to focus on various aspects of the input sequence. This enables the model to capture the intricate relationships and dependencies observed in the data.

As a result of its attention mechanism, the Transformer model performs better than traditional architectures in tasks like machine translation, and it can process long sequences and capture dependencies across elements. The Transformer design has become a mainstay of contemporary deep learning models by utilizing self-attention and multi-head methods, providing excellent performance and scalability in a range of NLP applications.

3.3.4 Position-wise Feed-Forward Networks

In addition to attention sub-layers, every layer in the encoder and decoder has a fully connected feed-forward network applied to each position independently and in the same way. This comprises a ReLU activation placed between two linear transformations. From layer to layer, the linear transformations use different parameters, even though they are the same across different positions. This can also be expressed as two convolutions with a kernel size of 1. The inner layer has dimensions $d_{ff} = 2048$, while the input and output have dimensionality $d_{model} = 512$ (Vaswani *et al.*, 2017).

3.3.5 Embeddings and Softmax

Similarly to other sequence transduction models, the learned embeddings were used to convert the input tokens and output tokens to vectors of dimension d_{model}. Additionally, the usual learned linear transformation and softmax function were employed to convert the decoder output to predicted next-token probabilities.

3.3.6 Positional Encoding

Transformer does not have recurrence and convolution; to enable the model to utilize the sequence's order, information about the relative or absolute position of the tokens must be incorporated. For this purpose, "positional encodings" are included with the input embeddings at the bottoms of the encoder and decoder stacks. To allow the two

to be added together, the positional encodings and the embeddings share the same dimension model.

3.4 Text-To-Text Transfer Transformer Model (T5)

The fundamental concept behind the text-to-text transformer model is to approach any text processing task as a "text-to-text" problem. This means that the model views every input problem as requiring the transformation of text into new text as output. By adopting this framework, it becomes possible to utilize the same model, objective, training procedure, and decoding process for each task under consideration (Raffel *et al.*, 2020).

3.4.1 T5 Architecture

As mentioned in the Transformer section above, the primary component of the Transformer model is self-attention, which replaces each element in a sequence with a weighted average of the rest of the sequence. The vanilla Transformer architecture comprised an encoder-decoder structure for sequence-to-sequence tasks but has evolved to include single Transformer layer stacks for various tasks like language modeling, classification, and span prediction. The encoder-decoder Transformer implementation follows the original proposal, with input tokens mapped to embeddings and passed through a stack of blocks containing self-attention layers and feed-forward networks. Layer normalization and residual skip connections are applied within each subcomponent of the encoder and decoder. The decoder includes standard attention mechanisms and autoregressive self-attention, allowing it to attend only to past outputs. Position signals are provided to the Transformer to make self-attention order-sensitive, with relative position embeddings being commonly used. The text-to-text transformer model largely aligns with the original transformer, but with changes such as removing layer norm bias, placing layer normalization outside the residual path, and using a different position

embedding scheme (Raffel *et al.,* 2020). In transfer learning, a model is pre-trained on a data-rich task before being fine-tuned on a downstream task of interest. This pre-training causes the model to develop general-purpose abilities and knowledge that can then be transferred to downstream tasks.

3.4.2 Pre-training Approach

The T5 model's pre-training approach includes careful selection of training data, meticulous tuning of hyperparameters, and any modifications that were made to improve the original transformer architecture's performance.

The pre-training phase leveraged a diverse and extensive dataset to facilitate robust learning of language representations. (Raffel *et al.,* 2020) used Common Crawl as their main source of text data for their study because it offers a large quantity of web-extracted material that has been scraped from the internet. However, the raw scraped text from a common crawl often includes non-natural language content such as gibberish, boilerplate text, offensive language, and code snippets. They used several cleaning heuristics to address this, such as removing pages that contained code snippets, removing pages that contained fewer than five sentences, filtering out offensive words, removing pages that mentioned JavaScript or contained lorem ipsum text, keeping only lines that ended in terminal punctuation marks, and deduplicating the dataset based on three-sentence spans. Additionally, they filtered out non-English content using language detection. While previous works have used common crawl data for NLP tasks, they chose to create a new dataset, named the "Colossal Clean Crawled Corpus" (C4), to provide a larger and cleaner text corpus. This dataset, obtained from web-extracted text in April 2019 and filtered according to their criteria, amounts to about 750 GB, and is made publicly available as part of the TensorFlow datasets.

As suggested by Vaswani *et al.* (2017), T5 adopts a straightforward encoder-decoder transformer architecture, which aligns well with the sequence-to-sequence nature of this task format. T5 is pre-trained using a "span-corruption" objective in masked language modeling, where consecutive spans of input tokens are replaced with a mask token. The model is then trained to reconstruct the masked-out token.

The "text-to-text" framework is an approach where the model receives input text and generates corresponding output text. This framework provides a consistent training objective for both pre-training and fine-tuning. In text classification tasks, the model easily predicts a single word corresponding to the target label. A diagram of the text-to-text framework with a few input/output examples is shown in Figure 4.

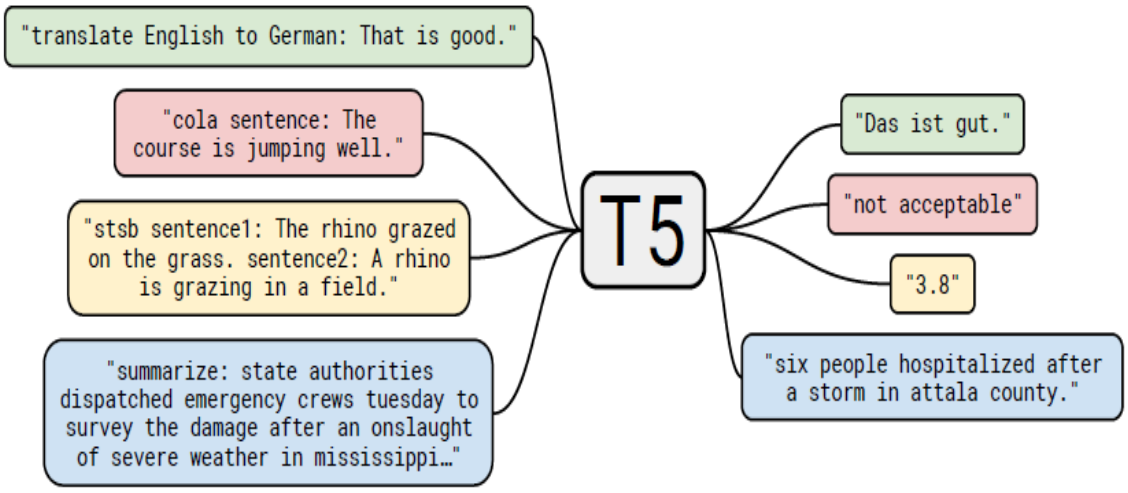

Figure 4. Text-to-Text (T5) Framework (Raffel *et al.*, 2020).

An extra advantage of T5 is its scale, with pre-trained model sizes available from 60 million to 11 billion parameters. 1 trillion tokens of data were used to pre-train these models. The large research study, extensively detailed by Raffel *et al.* (2020), served as the foundation for the selection of the pre-training aim, model architecture, scaling strategy, and numerous other design decisions made for T5.

The T5 model is pre-trained on the English language, and almost 80% of the world's population does not speak English. This limitation restricts the use of models that are pre-

trained solely on English-language text. Creating multilingual models pre-trained in a variety of languages is a more general methodology. mBERT, mBART, and XLM-R are popular models of this type; they are multilingual versions of BERT, BART, and RoBERTa, respectively. mT5, a multilingual variant of T5, is a model that deviates as little as possible from the recipe used to create T5. Because of this, mT5 has all the advantages of T5, including its adaptable text-to-text format and its design, which was informed by extensive empirical research (Xue *et al.*, 2020).

3.5 A Multilingual Text-to-Text Transformer (mT5)

The mT5 model, short for "multilingual T5," is a variant of the T5 model. It is designed to support various NLP tasks by framing them as sequence-to-sequence tasks, where text input is transformed into text output. mT5 leverages its multilingual capabilities to excel in tasks like classification, question answering, and machine translation.

The mT5 model architecture and training procedure closely follow those of T5. The mT5 is pre-trained on an unlabeled multilingual variant of the C4 dataset called mC4, created by Xue *et al.* (2020), which covers 101 languages. One of the crucial heuristic filtering steps in C4 was the removal of lines that did not end in an English terminal punctuation mark. The "line length filter" was utilized, requiring pages to have at least three lines of text with 200 characters or more, as many languages do not use English terminal punctuation marks. Pages with offensive language and duplicate lines across papers were also removed in accordance with C4's screening. Lastly, CLD3 was used to identify the predominant language of each page and eliminate any that had a confidence level lower than 70%. Following the application of these filters, the remaining pages were classified according to language, and those languages with 10,000 or more pages were added to the corpus.

The mT5 model follows the original T5 recipe and comes in five different sizes: small, base, large, XL, and XXL. The mT5 model is pre-trained for 1 million steps on batches of input sequences with a length of 1024, corresponding to approximately 1 trillion input tokens in total (Xue *et al.,* 2020).

3.6 Pre-trained Byte-to-Byte Model (ByT5)

The ByT5 model, introduced in a paper by Xue *et al.* (2022), is a token-free model designed to process text directly as raw bytes, eliminating the need for tokenization. ByT5 is based on the mT5 architecture and operates on UTF-8 bytes, offering benefits such as language agnosticism, robustness to noise, and reduced preprocessing complexity (Xue *et al.,* 2022).

Models such as BERT rely on a separate tokenization stage to divide documents into subword vocabulary. Consequently, there are more memory limits because massive embedding matrices with lots of parameters are needed for large vocabularies. Furthermore, the authors believe that sub-word tokenization is not robust in handling typos and lexical variants, and reducing all unknown words to the same out-of-vocabulary token prevents the model from distinguishing between different out-of-vocabulary (OOV) words.

As mentioned above, ByT5 is fed UTF-8 bytes without any preprocessing steps in place of words or subwords. The model only requires a dense matrix of 256 embeddings to represent byte-level embeddings, plus additional embeddings for the special tokens required to indicate the end of a sentence and pad sequences. Since every conceivable byte is represented in this instance, handling the OOV scenario is not necessary. Additionally, by eliminating the need for language-specific tokenization techniques, the model can be trained on multilingual corpora more easily according to the removal of the tokenization reliance. The "Span corruption" self-supervised objective is used to pre-train

the model, learning it to reconstruct 20-byte sequences that are masked in the input sentences (Gasparetto *et al.*, 2022).

While T5 and mT5 models use ‘‘balanced’ architectures, the encoder in the ByT5 model has a depth three times that of the decoder, as shown in Figure 5. ByT5, like the T5 and mT5 models, comes in different sizes (Small, Base, Large, XL, XXL) and is particularly effective for short-to-medium text sequences. In text classification benchmarks, ByT5 outperforms mT5 at the Small and Base sizes. The significant improvement in dense parameters over mT5 is likely the cause of ByT5's excellent performance at smaller scales, which is why we directly chose the small size of ByT5 in our proposed model.

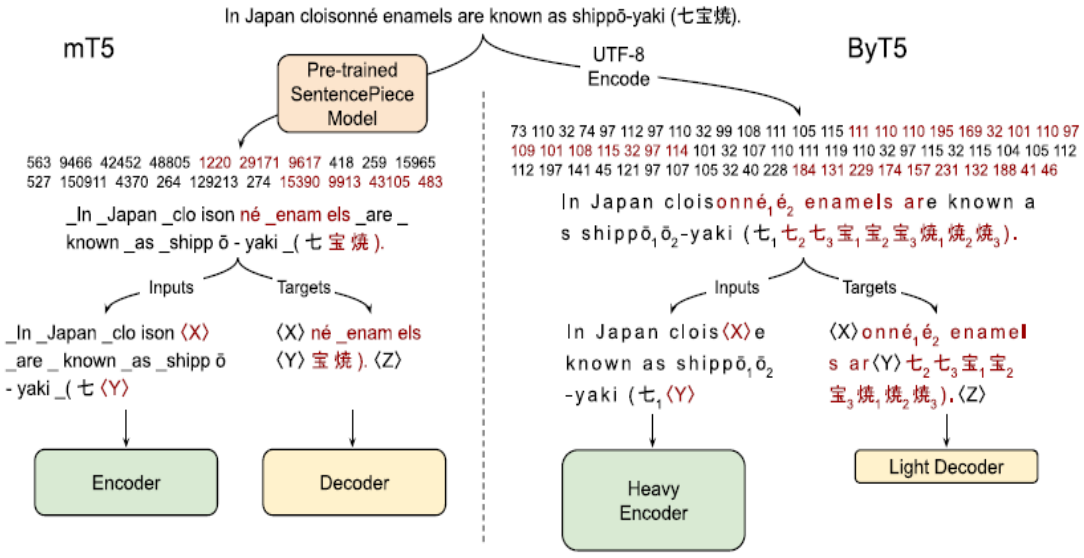

Figure 5. An Example Illustrates the Distinction between mT5 and ByT5 Architectures (Xue *et al.*, 2022).

Chapter Four

Datasets Preparation and Model Training

4.1 Introduction

In this chapter, we will outline the datasets utilized and the preparation steps undertaken to train the ByT5 model for Arabic Jordanian dialect identification. Subsequently, we will elucidate the training methodology employed in training the model. Finally, we will detail the evaluation metrics utilized to assess the performance of our model.

4.2 Datasets

We used in this work different dataset sources that are detailed in the following sections and subsections, along with some examples.

4.2.1 SDC Dataset

In this work, we used the SDC introduced by Abu Kwaik *et al.* (2018), who constructed the first Levantine Dialect Corpus. SDC is larger than the existing corpora in terms of size, words, and vocabularies; it is sourced from authentic conversations in social media and blogs, covering diverse topics; and it was created from scratch through two methods: automatic collection, using the Twitter API streaming library (Tweepy) to gather relevant tweets extensively; and manual collection, wherein sentences were extracted from the web, such as forums. SDC covers data from the four dialects spoken in Palestine, Jordan, Lebanon, and Syria, including several topics from regular conversations such as politics, education, society, health care, housekeeping, and others. To evaluate their corpus, they used a language identification task.

4.2.2 AOC Dataset

We used the AOC dataset by Zaidan and Callison-Burch (2011), who created a dataset of dialectal Arabic by extracting reader commentary from the online versions of three Arabic newspapers: Al-Ghad from Jordan, Al-Riyadh from Saudi Arabia, and Al-Youm Al-Sabe' from Egypt. Approximately half of the dataset comprises dialectal content. Zaidan and Callison-Burch conducted experiments involving automatic dialect classification, leveraging systems trained on the collected dialect labels, and presented their findings.

4.2.3 MADAR Dataset

Bouamor *et al.* (2018) presented the MADAR Corpus, which is a large parallel corpus of 25 Arabic city dialects in addition to the pre-existing parallel sets for English, French, and Modern Standard Arabic (MSA). These corpora are the first of their kind in terms of the breadth of their coverage and their precise details. They used the BTEC in French and English and translated 2,000 sentences into the different dialects. The translators are all native speakers of the dialects of the cities they descend from. We took from it 2,000 Jordanian dialect sentences. Also, we used a short Jordanian movie script named (ان شاء الله استفدت) with around 500 Jordanian sequences.

4.2.4 Combining the Different Datasets

We have assembled all the datasets related to the Jordanian dialect (JO) from the mentioned sources into one file. Additionally, we have obtained data on the Syrian dialect (SY) from the SDC dataset and on the Saudi Arabian dialect (SA), Egyptian dialect (EG), and MSA from the AOC dataset. Table 3 provides a glimpse of the dialects' dataset.

Table 3. Glimpse of Used Dataset.

Index	Text	Class
1	شو السالفه شو الموضوع	JO
2	ولما فتت لقيت الكل قاعدين وساكتين وما حدا عم يحكي سلمت عليهن وما حدا رد علي استغربت وفكرت نور و عباده متخانئين	SY
3	ما هو لو في حرية فى الدين مكنش ده كله حصل من اصله	EG
4	واذا بغيتو تتكلمون عن الممثلين جبتو شيفه والا عود يروع الجن	SA
5	وجمهور الوحدات كذلك	MSA

4.2.5 JODA Dataset

During the completion of writing and documenting this thesis, we obtained a dataset for the Jordanian dialect, which we have included in this section. The Jordanian Dataset (JODA) was collected, preprocessed, corrected, and labeled by the project team for the project "Developing Applications to Correct Jordanian Spoken Arabic to Proper Language Using Machine Learning Techniques." The samples in this dataset contain texts in the Jordanian dialect. The sources used for the dataset are classified into three categories: posts and comments from popular Jordanian social media pages and channels on Facebook, X (formerly Twitter), Instagram, and YouTube. The collected data primarily consisted of comments written by the audiences of these pages and channels, covering a wide range of economic, social, and political topics. Additionally, scripts from Jordanian movies and existing Arabic dialect datasets like SDC, AOC and MADAR were also used.

4.3 Dataset Preparation

The texts of different dialects contain duplicate sequences and unnecessary characters such as punctuation marks, numbers, and English letters. All these duplicate

sequences and characters were removed, then we combined all the datasets together, then shuffled the combined data, and finally we split it into a training set, a validation set, and a test set. Table 4 provides details of the dataset, including its sources, the total number of sequences both before and after preparation, the number of words, and the maximum sequence length in the dialect.

Table 4. Characteristics of Source Datasets.

Type of Texts	Dataset Source	Total Number of Sequences	Total Number of Sequences Used for Experiments	Number of words	Maximum Sequence Length
Jordanian dialect samples	AOC, SDC, MADAR and Short Jordanian film	20,954	20,530	348,219	1,015
Syrian dialect samples	SDC	37,760	4,966	67,870	836
Saudi Arabia dialect samples	AOC	20,741	4,966	64,180	495
Egyptian dialect samples	AOC	12,527	4,965	117,257	1,006
MSA samples	AOC	18,947	4,966	106,657	1,010

We divided the dataset into three parts: 80% for training, 10% for validation, and 10% for testing. The training dataset is used to train a machine learning model, allowing it to learn patterns and relationships between features and labels. The validation set is used to tune hyperparameters and assess model performance during training, helping to prevent overfitting. whereas the testing dataset serves as an independent dataset to evaluate the final model's performance, providing an unbiased assessment of its generalization ability to unseen data.

A standard test set was selected for all experiments we conducted, comprising predominantly Jordanian dialects, with the remaining half distributed equally among Saudi Arabian, Egyptian, and Syrian dialects, in addition to Modern Standard Arabic. We

ensured that test set texts did not overlap or duplicate with the training and validation sets. Table 5 illustrates the details of the data splits used in the experiments.

Table 5. Dataset Used in This Work (Max. Sequence Length = 1,015).

Dialect	Number of Sequences	Train Set	Validation Set	Test Set
JO	20,530	32,314	4,039	4,040
MSA	4,966			
SY	4,966			
SA	4,966			
EG	4,965			
Total Number of Sequences		40,393		

Table 6 provides details of JODA dataset, including its sources, the total number of sequences both before and after preparation, the number of words, and the maximum sequence length in the dialect.

Table 6. JODA Dataset Characteristics.

Type of Texts	Dataset Source	Total Number of Sequences	Total Number of Sequences Used for Experiments	Number of words	Maximum Sequence Length
Jordanian dialect samples	JODA	38,886	38,589	348,307	263
Syrian dialect samples	SDC	37,760	9,648	132,396	1,000
Saudi Arabia dialect samples	AOC	20,741	9,645	119,940	506
Egyptian dialect samples	AOC	12,527	9,648	235,862	1,022
MSA samples	AOC	18,947	9,648	191,521	1,010

Table 7 illustrates the details of the data splits used in the experiment which we used JODA dataset.

Table 7. Dataset Used to Improve This Work (Max. Sequence Length = 1,022).

Dialect	Number of Sequences	Train Set	Validation Set	Test Set
JO	38,589	70,942	3,118	3,118
MSA	9,648			
SY	9,648			
SA	9,645			
EG	9,648			
Total Number of Sequences		77,178		

4.4 Exploring ByT5 Model Selection and Hyperparameters

We used the small size of the ByT5 model, Table 8 illustrates the characteristics of small ByT5.

Table 8. Characteristics of Small ByT5

Size	Byt5/Small
Parameters	300 M
Vocab	0.3%
Model Dimension d_{model}	1,472
Feed Forward Dimension (d_{ff})	3,584
Encoder/Decoder Layers	12/4
Optimizer	AdaFactor
Learning Rate	0.003

Hyperparameters play a critical role in deep learning models as they greatly influence performance. Tuning these parameters is an essential aspect of the model-training process. In our study, we focused on adjusting the following hyperparameters:

1. **Maximum Sequence Length**: It denotes the maximum number of tokens allowed in a sample. Given that the ByT5 model operates on text directly as raw bytes, it is important to note that Arabic characters typically occupy 2 bytes each. Therefore, in

our thesis, we utilized two different maximum sequence lengths 2,048 and 1,024 in separate experiments to analyze their impact on the model's results.

2. **Batch Size**: It represents the number of training samples utilized in a single iteration of the training process before updating weights. It significantly impacts both the model's performance and the speed of convergence. In all our experiments, we maintained a train batch size of 256.
3. **The learning rate** (η) is a critical hyperparameter governing the rate at which an algorithm updates its parameter estimates. It directly influences the speed at which a neural network model learns about a problem. A properly calibrated learning rate is crucial for efficient learning, affecting both the convergence speed and the quality of the final weights. Through experimentation, we determined that a learning rate of 0.003 was more suitable and yielded higher accuracy compared to a learning rate of 0.001.
4. **Optimizer**: it is instrumental in improving the performance of neural networks by iteratively adjusting parameters, such as weights, to minimize losses. In our thesis, we conducted experiments with various optimizers and ultimately adopted the Adafactor optimizer. This decision followed experimentation with the Adam weight decay optimizer, which is based on the Adam algorithm.

4.5 Model Training

We utilize the pre-trained ByT5 model provided by Xue *et al.* (2022), which is originally based on the mT5 model (Xue *et al.,* 2021), trained on the mC4 dataset.

For fine-tuning the ByT5 checkpoints, we adhere to the procedure outlined in Raffel *et al.* (2019) without any additional hyperparameter tuning. Specifically, we employ the AdaFactor optimizer (Shazeer and Stern, 2018) with a learning rate of 0.003, along with dropout during training, to enhance the model's performance. Additionally,

we determined a batch size of 256 through experimentation, which facilitates faster training and improved outcomes.

During evaluation, we reserve 10% of the training set for validation, fine-tuning the model with the remaining 90% of examples. The best-performing checkpoint is selected for final evaluation on the test set. We train the model for 1,500 steps, observing that validation accuracy typically plateaus after 1,000 steps without showing signs of overfitting.

To adapt text classification tasks to the text-to-text format, we input the text or sequence into the model and train it to predict the literal class text as output. In our case, the model predicts the Jordanian dialect among various Arabic dialects, including MSA.

4.6 Experiments Platforms

This section describes the experimental platforms used in dataset preparation, training the ByT5 model, conducting experiments to enhance it, and obtaining results. The experimental platform utilized in this thesis is Colab Pro Plus. Colab Pro Plus was employed to train the proposed model on the entire dataset simultaneously, thanks to its high-capacity resources such as random-access memory (RAM) and GPU RAM. Table 9 displays the specifications of the Colab Pro Plus platform. We stored the datasets and the ByT5 model configuration files in Google Cloud Storage (GCS). Additionally, the results of the experiments were stored every 100 steps. Google Cloud Storage (GCS) provides a reliable, scalable, and high-performance solution for storing datasets and model configuration files.

Table 9. Specifications of Colab Pro Plus Platform

Components	specification
CPU	Intel(R) Xeon(R) CPU @ 2.20GHz, 4 cores, 56 MB cache
TPU	v2
GPU	Nvidia-Tesla V100-SXM2 @ 1.32GHz, 16 GB Nvidia-Tesla P100-PCIE @ 1.32GHz, 16 GB
Memory	53 GB
Runtime Limit	Sessions limited to 24 hours
Libraries	Python 3.7.13, TensorFlow 2.12.0 Tensorboard 2.12.0, Tensor2tensor 1.15.7 Jax 0.4.25, NumPy 1.23.5 Pandas 1.5.3, Datasets 2.5.0

4.7 Evaluation Metrics

We evaluate our model using the following metrics: accuracy and F1 score, which are the most used metrics in language classification.

4.7.1 Accuracy

Accuracy is a fundamental evaluation metric in language classification, measuring the model's ability to make correct predictions. It gauges overall correctness by calculating the ratio of correctly predicted instances to the total instances. A model can produce four types of predictions: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Accuracy is simply the ratio of correctly predicted observations to the total observations. It is calculated using Equation 2.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

4.7.2 F1 Score

The F1 score is an important metric in language classification, providing a fair assessment of the model's performance by combining precision and recall into a single measure. It is calculated in Equation 5 as the harmonic mean of precision and recall, offering insights into how well the model balances between correctly identifying positive instances (precision) and capturing all positive instances (recall), precision and recall are calculated in Equations 3 and 4 respectively. The F1 score is particularly useful in

scenarios where both precision and recall need to be optimized simultaneously, such as in Arabic dialect classification, where achieving a balance between correctly identifying specific dialects and capturing all instances of those dialects is essential for accurate classification. By considering both precision and recall, the F1 score is a useful metric to evaluate language classification models since it offers a thorough evaluation of the model's performance, particularly when it relates to Arabic dialect classification.

$$Precision = \frac{TP}{TP+FP} = \frac{true\ positive}{total\ predicted\ positive} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} = \frac{true\ positive}{total\ actual\ positive} \quad (4)$$

$$F1\_Score = \frac{2 \times (Precision \times Recall)}{Precision+Recall} \quad (5)$$

Chapter Five

Experiments and Results

5.1 Introduction

In this chapter, we will discuss the experiments and trials we conducted to identify the Jordanian dialect texts. We used accuracy and the F1 score to show the models' ability to predict the Jordanian dialect between various dialects.

5.2 Limited Availability of Data

Since there is no dedicated database for the Jordanian dialect and there is a need to accommodate the dialectal diversity across Jordan's governorates at the time of writing this thesis, we have gathered Jordanian dialect data from various sources cited in Chapter 4 into a single database. Following thorough cleaning and preparation, we possess approximately twenty thousand sequences that are primed for experimentation.

5.3 Jordanian Dialect Identification Model

As mentioned in section 4.4 we utilize the pre-trained ByT5 model provided by Xue *et al*. (2022), to make several experiments to obtain best result in identify the Jordanian dialect. We formulate our Jordanian dialect identification problem as a multi-class classification task. The AdaFactor optimizer with a learning rate of 0.003 and a batch size of 256 are used in training the model. Figure 6 shows the best result achieved using ByT5 model, it achieved 88.7% F1 score.

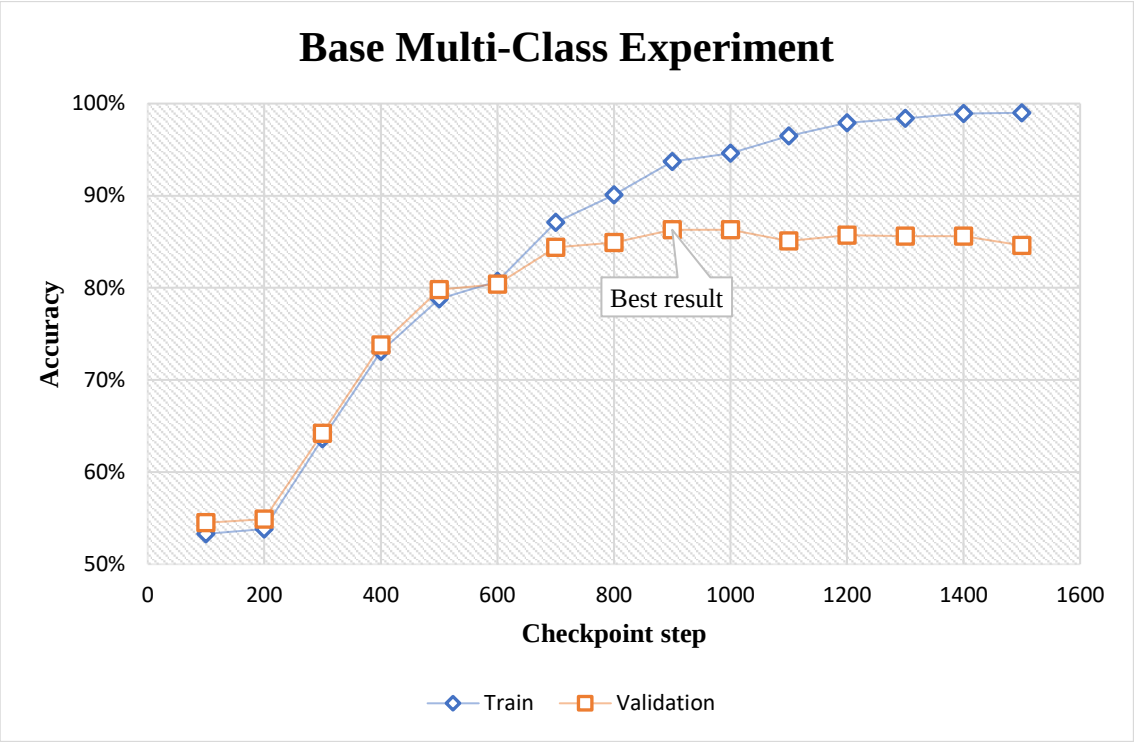

Figure 6. Performance of ByT5 Model in Jordanian Dialect Identification.

5.4 Training Experiments

In our study, we conducted two basic experiments, each focusing on assessing the performance of the ByT5 model. The primary experiment that achieved the highest accuracy and F1 score, termed the 5-way experiment, involved multi-class classification tasks, where the model was tasked with distinguishing between JO dialect and various Arabic dialects along with MSA sentences, comprising five classes as detailed in Chapter 4. The training, validation, and test data for these classes are presented in Table 5 Additionally, the remaining experiment was categorized as 2-way experiment, involving binary classification task to differentiate between JO as the main class and SA, EG, SY dialects, and MSA as one class identified as non-Jordanian.

The AdaFactor optimizer with a learning rate of 0.003 and a batch size of 256 are used in training the model across all experiments. Table 10 shows the results of the main experiments that we conducted, indicating that the experiment that obtained the highest result was the 5-way experiment, which we will refer to as the “base multi-class

experiment”, and we will make several changes in the hyperparameters and using larger dataset to indicate whether its results improve.

Table 10. Performance Comparison of Byt5 Model on Various Experiment.

Model	Dataset Dialects	Number of Classes	Validation Set		Test Set	
			Accuracy	F1 Score	Accuracy	F1 Score
Small ByT5	JO and non-JO	2	85.4%	86.4%	85.7%	86.3%
Small ByT5	JO, SY, SA, MSA and EG	5	86.3%	89.1%	86.0%	88.7%

In the following sections, we will discuss, present, and analyze the results of the base multi-class experiment by changing the sequence length, learning rate, and type of optimizer. Additionally, we will use the JODA dataset, which is nearly twice the size of the dataset initially used.

5.4.1 Different Sequence Length

This experiment aimed to test the effect of changing the sequence length on the model's performance in the "base multi-class experiment", which initially used a maximum sequence length of 2048. We reduced the sequence length of the input texts to less than 1024 bytes, noting that Arabic characters typically occupy 2 bytes each. Figure 7 illustrates the lack of improvement in the results of the "base multi-class classification" when reducing the length of the input texts. This suggests a certain degree of similarity between the entered dialect classes and indicates that the model requires longer sentences to determine the dialect more accurately.

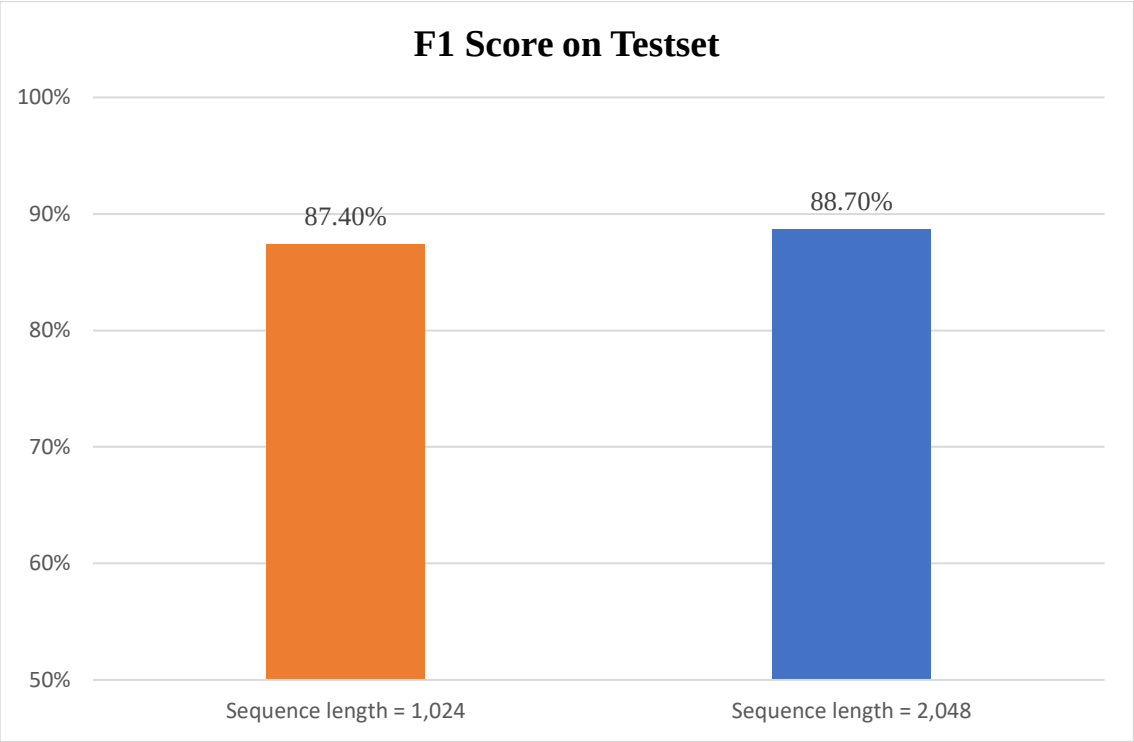

Figure 7. Impact of Decreasing Sequence Length on Base Multi-Class Experiment.

5.4.2 Changing the Learning Rate

In the pursuit of optimizing model performance in Jordanian dialect identification, experimentation with hyperparameters such as the learning rate proved essential. We explored the impact of changing the learning rate from 0.003, which the "base multi-class experiment" trained on, to 0.001. Contrary to expectations, this adjustment did not yield improved results; instead, it led to a decrease in performance, as depicted in Figure 8. This observation underscores the importance of careful hyperparameter tuning and highlights the intricate balance required to achieve optimal model performance.

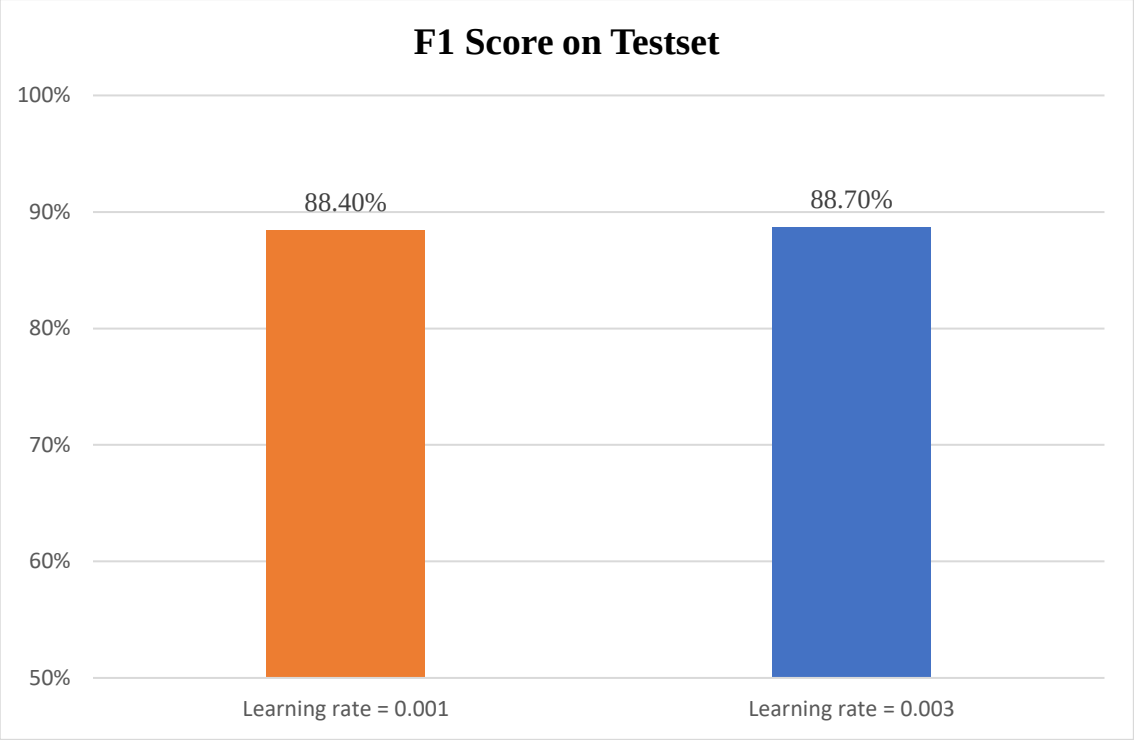

Figure 8. Impact of Learning Rate Variation on Base Multi-Class Experiment.

5.4.3 Using Different Optimizer

In fact, the aim of optimizers is to update the attributes of neural networks, like weights and biases, during training to reduce losses, so we conducted an experiment involving the transition from the AdaFactor optimizer to Adam with weight decay on "base multi-class experiment". The switch in optimizers did not lead to improved performance, as depicted in Figure 9. This unexpected result prompts a reevaluation of the influence of optimization algorithms on our model's efficacy. It underscores the nuanced interplay between optimizer selection and model architecture, suggesting that while certain optimizers may be well-suited for specific tasks, their effectiveness can vary depending on the intricacies of the dataset and model configuration.

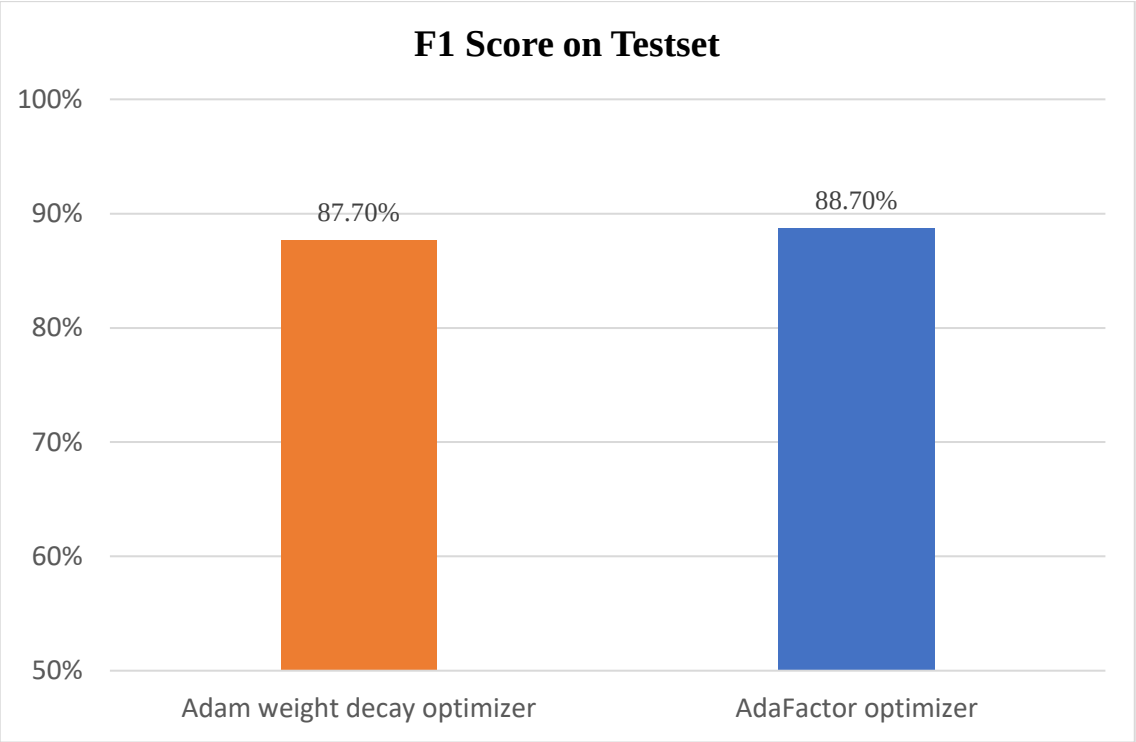

Figure 9. Impact of Changing the Optimizer on Base Multi-Class Experiment.

5.4.4 Using JODA Dataset

By analyzing vast amounts of data, deep learning models develop the ability to recognize patterns. The basic concept is that with more input, the model learns more varied and complex representations, which improves performance. A larger dataset provides the model with additional examples to learn from, reducing the chance of overfitting and enhancing its ability to generalize to unseen data. In this experiment, we used the JODA dataset, which contains approximately 77k sequences. The impact of increasing the dataset was very positive, as demonstrated in Figure 10, where the F1 score improved to 95.3% on test set at 1400 checkpoint step.

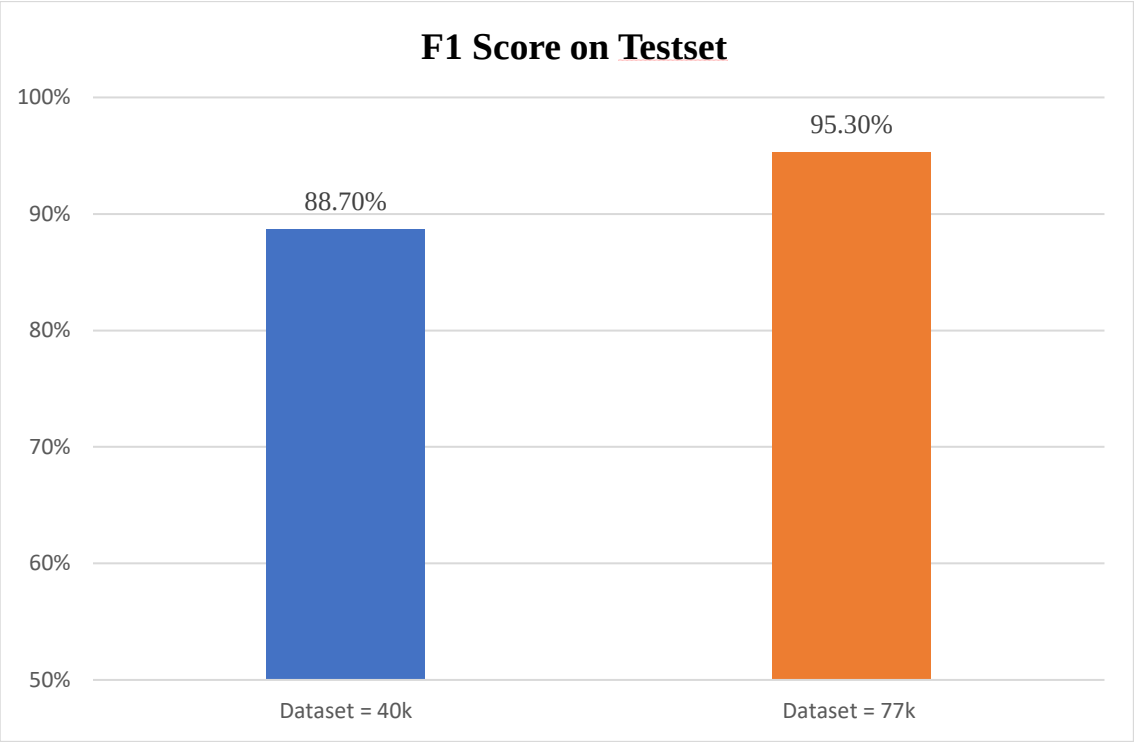

Figure 10. Impact of Using JODA Dataset on Base Multi-Class Experiment.

5.5 Model Timing

Across most of our experiments, we observed consistent timing, indicative of stable computational performance. However, a notable exception arose when we altered the sequence length parameter. Specifically, experiments involving variations in sequence length exhibited discernible discrepancies in training times. For instance, experiment with shorter sequence lengths demonstrated reduced training times compared to those with longer sequences. It requires 7.67 seconds per step, resulting in a total training time of 3.19 hours (including 1,500 steps), and takes 1.18 seconds to classify each test sequence. Although the base multi-class experiment requires 7.97 seconds per step, resulting in a total training time of 3.32 hours (including 1,500 steps), it maintains an acceptable response time in prediction mode. Specifically, it takes 1.48 seconds to classify each test sequence.

5.6 Analysis and Discussion

All Arabic dialects utilize the same character set, and moreover, a significant portion of the vocabulary is shared among different dialects. Thus, distinguishing between the dialects and recognizing the Jordanian dialect specifically is not a straightforward endeavor. Table 11 illustrates some of the sentences that the Byt5 model classified correctly and incorrectly.

Table 11. Examples of correctly and incorrectly classified Sentences.

Texts	Target	Prediction	Result
إنت فاهم شو يعني إنه الكل بحاول يلفت انتباهي وأنا مش منتبهه غير إلك	JO	JO	Correct
انعام ليش زعجتك بشي	SY	SY	Correct
هي عزبة ملا إيه	EG	EG	Correct
القريني شاهد ماشفش حاقه ماراح يفوزون	SA	SA	Correct
الحكم بالوقت الضائع والنتيجة	MSA	MSA	Correct
عفواً ممكن أسألك سؤال	MSA	JO	Incorrect
فيها توعية فيها تثقيف فيها متعة	SA	JO	Incorrect
ضحك علي	JO	SA	Incorrect

We have made numerous efforts to improve the results of the base multi-class experiment. These efforts included adjustments such as modifying the learning rate, changing the optimizer, reducing the input text sequence length, and expanding the dataset by using the JODA dataset. The use of the JODA dataset led to a significant improvement, achieving a 95.3% F1 score instead of 88.7% for the base multi-class experiment. Based on the experiments we conducted on the base multi-class experiment, we determined that the optimal learning rate is 0.003, and the preferred optimizer is AdaFactor with a 256-train batch size.

The following confusion matrix illustrates the model's effectiveness in identifying Jordanian dialect text, achieving a 91.53% success rate. However, it did make some mistakes, with misclassifications occurring as follows: 2.69% misclassified as SA, 3.64% misclassified as SY, 0.45% misclassified as EG, and 1.86% misclassified as MSA. The high values on the diagonal (from top left to bottom right) indicate that the model

generally performs well in classifying each dialect correctly. The model is highly accurate in predicting Syrian Arabic texts, with almost all instances correctly classified.

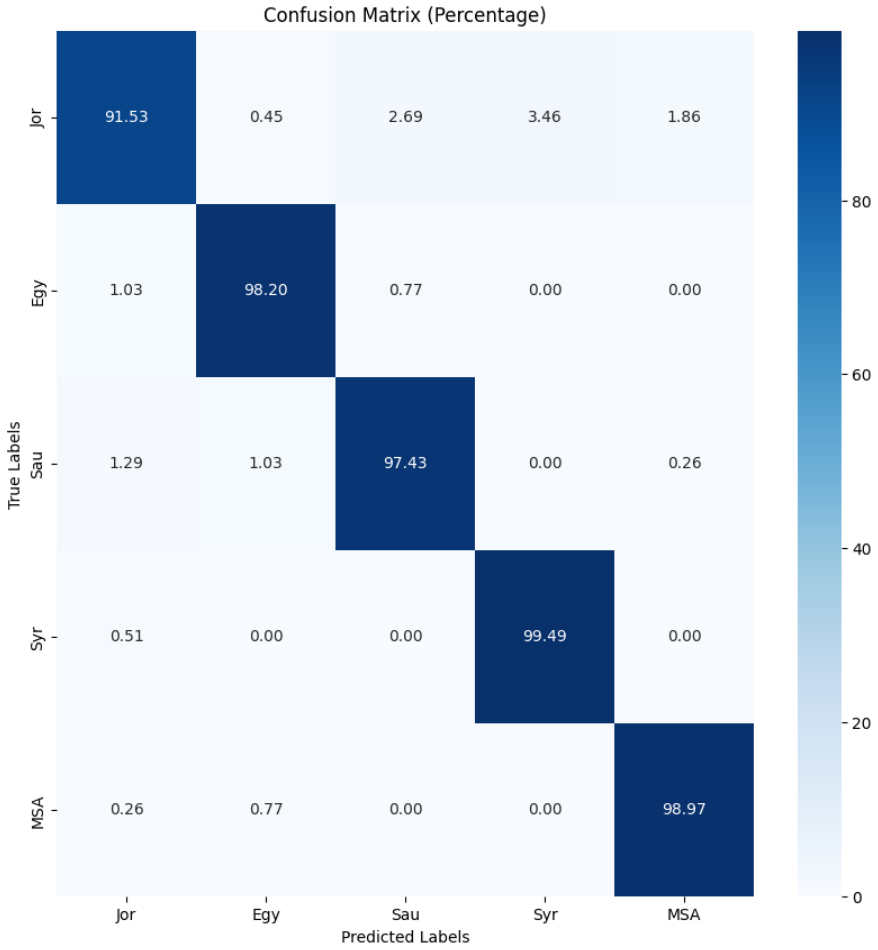

Figure 11. Confusion Matrix Analysis for Jordanian Dialect Identification.

The main question posed in this thesis was: which ML algorithm is suitable for the ADI to recognize Jordanian Arabic dialect?

As mentioned in the earlier chapters of the thesis, this issue is not easy due to several reasons, the most important of which is the significant overlap between dialects and their sharing of many vocabulary items. Additionally, some of these vocabulary items and words may resemble each other in writing but differ in pronunciation. After research, reading, and comparison, we found it appropriate to use deep learning models, especially those based on transformers such as the T5 model and its extensions like Byt5.

Transformer-based models like T5 have demonstrated remarkable effectiveness in text classification tasks due to their unified text-to-text framework, pre-training tasks, and

size variants. T5 processes all tasks, including translation, summarization, question answering, and text classification, in the same manner by converting them into a text-to-text format. T5 is an extremely adaptable NLP tool because of its unified approach, which enables it to adapt to almost any text-based behavior. T5 is pre-trained using unsupervised learning on a variety of text corpuses, including word masking and word shuffles, to improve the model's understanding of the context and structure of language. With this pre-training, T5 is more adept at text classification tasks since it can identify patterns and correlations in language.

The token-free nature is a key feature of the ByT5 model, making it efficient and beneficial for text categorization tasks in NLP. Compared to token-based models like mT5, ByT5 offers significant advantages as a state-of-the-art token-free Transformer architecture. ByT5's generative capabilities benefit from its concise vocabulary and minimal parameter usage, allowing for the allocation of additional parameters to other aspects. Additionally, ByT5's architecture consumes less memory than mT5's, contributing to its performance in NLP text classification tasks. ByT5's token-free approach and efficient memory utilization play a crucial role in its performance. Moreover, ByT5's ability to process text in any language without requiring language-specific tokenizers simplifies the language classification process by eliminating the need for separate tokenizers for each language.

To overcome the lack of training data for the Jordanian dialect, we searched for dialect resources on the Internet and found several research papers containing texts in the dialect. These were referenced in Chapter Four of this thesis. Recently, we used the JODA dataset, which made a positive difference in the results of the base multi-class experiment.

Increasing the dataset for deep learning models for Jordanian dialect detection can improve their robustness. A larger dataset allows the model to learn a more diverse range of patterns and features, leading to better generalization and improved performance on unseen data. This is because the model has more examples to learn from, helping it capture the complexity and variability of natural language.

Chapter Six

Conclusion and Future Work

In recent years, transformer models have proven highly effective in various NLP tasks. In this thesis, we utilize a DL ByT5 model for identifying Jordanian dialects among various others. The model achieves an impressive accuracy rate in identifying the Jordanian dialect, boasting an F1 score of 95.3% in multi-class classification tasks. The model distinguishes between the JO dialect and various Arabic dialects, along with MSA sentences. To train our proposed model, we utilized a dataset comprising 77K sequences from diverse sources.

While the DL ByT5 model demonstrates commendable accuracy in identifying the Jordanian dialect, several avenues for future exploration emerge, critical for advancing its capabilities and applicability in real-world scenarios. Here are some potential areas for future work:

- Fine-tuning on a Larger Dialect Corpus: Increasing the size and diversity of the Jordanian dialect corpus used for training could significantly enhance the model's performance. By incorporating a larger and more varied dataset, the model could better capture the nuances and variations within the Jordanian dialect, leading to improved specificity and generalization. This endeavor may involve collecting additional textual data from sources such as social media, online forums, news articles, and literature.
- Multimodal Fusion Methods: Expanding the model's capabilities to process multimodal inputs, such as integrating audio or video data alongside text, could facilitate a more comprehensive understanding of language. Exploring multimodal fusion methods, such as attention mechanisms that dynamically integrate information from different modalities, could leverage complementary cues from multiple sources.

For instance, combining textual data with audio recordings of spoken dialects or videos depicting cultural contexts could provide richer contextual information for dialect identification.

- Cross-Dialect Analysis: Extending the study's scope to include a comparative analysis of Jordanian dialects with other Arabic dialects or languages could yield valuable insights into linguistic variations and similarities. Analyzing how the DL ByT5 model performs in distinguishing between different dialect pairs would offer a deeper understanding of the unique features and challenges associated with each dialect.

Additionally, leveraging Large Language Models (LLMs), such as GPT-3, for Jordanian dialect identification presents a promising avenue for further research and development. LLMs excel in capturing intricate linguistic patterns and contextual nuances from large-scale text corpora through unsupervised pre-training, making them well-suited for dialect identification tasks. By fine-tuning LLMs on a Jordanian dialect dataset, researchers can harness their robust language understanding capabilities to discern subtle variations in dialectal expressions and vocabulary, thereby advancing the field of dialect identification.

References

Abandah, G., Khedher, M., Anati, W., Zghoul, A., Ababneh, S., & Hattab, M. S. (2015, May). The Arabic language status in the Jordanian social networking and mobile phone communications. In **7th Int'l Conference on Information Technology (ICIT 2015**) (pp. 449-456).

Abdel-Salam, R. (2022, December). Dialect & sentiment identification in nuanced Arabic tweets using an ensemble of prompt-based, fine-tuned, and multitask BERT-based models. In **Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)** (pp. 452-457).

Abdul-Mageed, M., Elmadany, A., Zhang, C., Nagoudi, E. M. B., Bouamor, H., & Habash, N. (2023). NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task. **arXiv preprint arXiv**:2310.16117.

Abdul-Mageed, M., Zhang, C., Bouamor, H., & Habash, N. (2020, December). NADI 2020: The First Nuanced Arabic Dialect Identification Shared Task. **In Proceedings of the Fifth Arabic Natural Language Processing Workshop** (pp. 97-110). Association for Computational Linguistics.

Algihab, W., Alawwad, N., Aldawish, A., & AlHumoud, S. (2019). Arabic speech recognition with deep learning: A review. In **Social Computing and Social Media. Design, Human Behavior and Analytics: 11th International Conference, SCSM 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26-31, 2019, Proceedings, Part I 21** (pp. 15-31). **Springer International Publishing**.

Alkhamissi, B., Gabr, M., ElNokrashy, M., & Essam, K. (2021, April). Adapting MARBERT for Improved Arabic Dialect Identification: Submission to the NADI

2021 Shared Task. **In Proceedings of the Sixth Arabic Natural Language Processing Workshop** (pp. 260-264).

Althobaiti, M. J. (2020). Automatic Arabic dialect identification systems for written texts: A survey. **arXiv preprint arXiv**:2009.12622.

Bouamor, H., Habash, N., Salameh, M., Zaghouani, W., Rambow, O., Abdulrahim, D., ... & Oflazer, K. (2018, May). The madar Arabic dialect corpus and lexicon. In **Proceedings of the eleventh international conference on language resources and evaluation** (LREC 2018).

Bouamor, H., Hassan, S., & Habash, N. (2019, August). The MADAR shared task on Arabic fine-grained dialect identification. **In Proceedings of the Fourth Arabic Natural Language Processing Workshop** (pp. 199-207).

Colin, R. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. **JMLR**, 21(140), 1.

Colin, R. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. **JMLR**, 21(140), 1.

Demszky, D., Sharma, D., Clark, J., Prabhakaran, V., & Eisenstein, J. (2021, June). Learning to Recognize Dialect Features. **In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics**.

El Mekki, A., El Mahdaouy, A., Essefar, K., El Mamoun, N., Berrada, I., & Khoumsi, A. (2021, April). BERT-based multi-task model for country and province level MSA and dialectal Arabic identification. In **Proceedings of the sixth Arabic natural language processing workshop** (pp. 271-275).

Elaraby, M., & Abdul-Mageed, M. (2018, August). Deep models for Arabic dialect identification on benchmarked data. In **Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)** (pp. 263-274).

Elaraby, M., & Abdul-Mageed, M. (2018, August). Deep models for Arabic dialect identification on benchmarked data. In **Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)** (pp. 263-274).

El-karef, M., Moses, M., Tanaka, S., Barry, J., & Mel, G. (2023, December). NLPeople at NADI 2023 Shared Task: Arabic Dialect Identification with Augmented Context and Multi-Stage Tuning. In **Proceedings of ArabicNLP 2023** (pp. 642-646).

Elnagar, A., Al-Debsi, R., & Einea, O. (2020). Arabic text classification using deep learning models. **Information Processing & Management**, 57(1), 102121.

Elnagar, A., Yagi, S. M., Nassif, A. B., Shahin, I., & Salloum, S. A. (2021). Systematic literature review of dialectal Arabic: identification and detection. **IEEE Access**, 9, 31010-31042.

Fares, Y., El-Zanaty, Z., Abdel-Salam, K., Ezzeldin, M., Mohamed, A., El-Awaad, K., & Torki, M. (2019, August). Arabic dialect identification with deep learning and hybrid frequency based features. In **Proceedings of the Fourth Arabic Natural Language Processing Workshop** (pp. 224-228).

Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). A survey on text classification algorithms: From text to predictions. **Information,** 13(2), 83.

Harrat, S., Meftouh, K., Abidi, K., & Smaïli, K. (2019). Automatic identification methods on a corpus of twenty five fine-grained Arabic dialects. In **Arabic Language Processing: From Theory to Practice: 7th International Conference, ICALP**

2019, Nancy, France, October 16–17, 2019, Proceedings 7 (pp. 79-92). Springer International Publishing.

Islam, S., Elmekki, H., Elsebai, A., Bentahar, J., Drawel, N., Rjoub, G., & Pedrycz, W. (2023). A comprehensive survey on applications of transformers for deep learning tasks. **Expert Systems with Applications**, 122666.

Kwaik, K. A., Saad, M., Chatzikyriakidis, S., & Dobnik, S. (2018, May). Shami: A corpus of levantine Arabic dialects. In **Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018**).

Kwaik, Kathrein Abu, and Motaz Saad. "ArbDialectID at MADAR shared task 1: Language modelling and ensemble learning for fine grained Arabic dialect identification**." Proceedings of the Fourth Arabic Natural Language Processing Workshop.** 2019.

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., ... & He, L. (2022). A survey on text classification: From traditional to deep learning. **ACM Transactions on Intelligent Systems and Technology (TIST**), 13(2), 1-41.

Lulu, L., & Elnagar, A. (2018). Automatic Arabic dialect classification using deep learning models. **Procedia computer science**, 142, 262-269.

Luo, X. (2021). Efficient English text classification using selected machine learning techniques. **Alexandria Engineering Journal,** 60(3), 3401-3409.

Mageed, M. A., Zhang, C., Elmadany, A., Bouamor, H., & Habash, N. (2021, April). NADI 2021: The Second Nuanced Arabic Dialect Identification Shared Task. **In Proceedings of the Sixth Arabic Natural Language Processing Workshop** (pp. 244-259).

Mageed, M. A., Zhang, C., Elmadany, A., Bouamor, H., & Habash, N. (2022, December). NADI 2022: The Third Nuanced Arabic Dialect Identification Shared Task.

In Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP) (pp. 85-97).

Salameh, M., Bouamor, H., & Habash, N. (2018, August). Fine-grained Arabic dialect identification. In **Proceedings of the 27th international conference on computational linguistics** (pp. 1332-1344).

Shazeer, N., & Stern, M. (2018, July). Adafactor: Adaptive learning rates with sublinear memory cost. **In International Conference on Machine Learning** (pp. 4596-4604). PMLR.

Stankevičius, L., Lukoševičius, M., Kapočiūtė-Dzikienė, J., Briedienė, M., & Krilavičius, T. (2022). Correcting diacritics and typos with a ByT5 transformer model. **Applied Sciences**, 12(5), 2636.

Talafha, B., Ali, M., Za'ter, M. E., Seelawi, H., Tuffaha, I., Samir, M., ... & Al-Natsheh, H. (2020, December). Multi-dialect Arabic BERT for Country-level Dialect Identification. **In Proceedings of the Fifth Arabic Natural Language Processing Workshop** (pp. 111-118).

Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient transformers: A survey. **ACM Computing Surveys**, *55*(6), 1-28.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. **Advances in neural information processing systems**, 30.

Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., ... & Raffel, C. (2022). Byt5: Towards a token-free future with pre-trained byte-to-byte models. **Transactions of the Association for Computational Linguistics**, 10, 291-306.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., ... & Raffel, C. (2021). mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. **In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics.**

Zaidan, O. F., & Callison-Burch, C. (2014). Arabic dialect identification. **Computational Linguistics**, 40(1), 171-202.

Zaidan, O., & Callison-Burch, C. (2011, June). The Arabic online commentary dataset: an annotated dataset of informal Arabic with high dialectal content. In **Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies** (pp. 37-41).

Zhang, C., & Abdul-Mageed, M. (2019, August). No army, no navy: Bert semi-supervised learning of Arabic dialects. In **Proceedings of the Fourth Arabic Natural Language Processing Workshop** (pp. 279-284).

التعرف على نصوص باللهجة العربية الأردنية باستخدام تقنيات تعلم الآلة

إعداد

الاء محمد مصلح السواعير

المشرف

الأستاذ الدكتور غيث علي عبندة

ملخص

نتيجة للثورة في وسائل التواصل الاجتماعي في العقد الأخير، بدأت اللهجات العربية العامية، بالظهور مكتوبة. تشير "اللهجة" إلى مجموعة من الهياكل اللغوية والمفردات الخاصة ببيئة معينة. وتختلف اللهجات أيضًا في نطق بعض الكلمات، مما يميز مجموعة من الأفراد عن غيرهم. بين اللهجات العربية الأصيلة، تبرز اللهجة الأردنية، التي تُعتبر إحدى اللهجات الشامية، والتي تتفرع أيضًا إلى تقسيمات إضافية تبعًا للمناطق الجغرافية. يعد التعرف على اللهجات العربية عنصرًا أساسيًا لمهام معالجة اللغات الطبيعية، بما في ذلك تحليل نصوص وسائل التواصل الاجتماعي والترجمة الآلية. يمكن اعتبار تحدي تحديد اللهجة أصعب من تحدي تحديد اللغة نفسها نظرًا لأن اللهجات يمكن اعتبارها لغات ذات صلة وثيقة بعضها ببعض، لذا يظل تحديد اللهجة آليًا لنص عربي تحديًا كبيرًا للباحثين. تمتلك تقنية المحولات المعاصرة تأثيرًا كبيرًا ضمن إطار التعلم العميق في الوقت الحالي. حيث تستخدم المحولات على نطاق واسع ولها نجاحات كبيرة في العديد من المجالات، وتلعب دورًا هامًا في مهام معالجة اللغات الطبيعية، بما في ذلك تحديد اللغة. يعتمد نموذج حرف-إلى-حرف (ByT5) المدرب مسبقًا على معمارية T5، وهو نموذج من شركة جوجل من نماذج المحولات الناجحة. يُعتبر ByT5 نموذجًا لا يعتمد على ترميز الكلمات المعالجة بل يقبل المُدخل بدون ترميز مباشرة كبايتات خام، مما يلغي الحاجة إلى رمزة النص، ويعمل على ترميز UTF-8 للأحرف، مما يوفر فوائد عدة مثل عدم التحيز للغة والصلابة أمام الضوضاء وتقليل تعقيد المعالجة المسبقة. تم جمع نصوص اللهجة الأردنية من مصادر متعددة للبيانات مثل التعليقات العربية عبر الإنترنت (AOC)، ومجموعة لهجات الشام (SDC)، ومشروع التطبيقات والموارد المتعددة للهجات العربية (MADAR)، بإجمالي حوالي 20 ألف نص باللهجة الأردنية. قمنا بإجراء عدة تجارب باستخدام نموذج التعلم العميق

ByT5 لتحدي التعرف على اللهجة الأردنية بين اللهجات العربية المختلفة. حقق النموذج دقة جيدة جدًا في تحديد اللهجة الأردنية، حيث حصل على درجة F1 بلغت 88.7% من تجربة التصنيف متعدد الفئات. بالإضافة إلى ذلك، قمنا بتحسين هذه النتيجة باستخدام مجموعة بيانات اللهجة الأردنية (JODA)، التي تحتوي على حوالي 39 ألف تسلسل من اللهجة الأردنية. وقد أدى مضاعفة حجم مجموعة بيانات اللهجة الأردنية من 20 ألف إلى 39 ألف تسلسل إلى تحسين درجة F1 للكشف عن اللهجة الأردنية من 88.7% إلى 95.3%.