

مروه المحمود

Master Thesis Template-10-2-2026.docx

 Midocean University

Document Details

Submission ID

trn:oid:::2945:404793446

Submission Date

Aug 5, 2026, 9:50 PM GMT+3

Download Date

Aug 9, 2026, 4:15 AM GMT+3

File Name

Master Thesis Template-10-2-2026.docx

File Size

658.8 KB

44 Pages

5,392 Words

34,565 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely human generated as AI generated and likely AI generated as human generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.



Title

Prepared By

Name: Marwa Mohammed AL-Mahmoud ID No: 261000175

Name: Norah Homoud Alharbi ID No: **222000773**

Name: Nouf Ayed Alalkabi ID No: **261002271**

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

Collage of Informatics

Midocean University

UAE

Supervisor Name: DR. Hajer Saleh

Collage of Informatics, Midocean University, Comoros

Email:.....

Acknowledgements

Special thanks to Dr. Hajer Saleh

Abstract

List of Publications

Table of Contents

List of Abbreviations	vii
List of Figures.....	viii
List of Tables	ix

Chapter 1: Introduction and

Overview.....	1
1.2 Problem Definition	2
1.3 Thesis Objectives	3
1.4 Thesis Contribution	4
1.5 Thesis Organization.....	5

Chapter 2: Background and Related Work

.....	31
-------	----

Chapter 3: Proposed Approach 78

4.1 Introduction	78
------------------------	----

Chapter 4: Experimental Results and Discussion 86

Chapter 5: Conclusion and Future Work 127

7.1 Important Features	127
7.3 Future Work	129

References	130
------------------	-----

List of Abbreviations

- **Acc.:** Accuracy
- **ANN:** Artificial Neural Network
- **ANOVA:** Analysis of Variance
- **ROC:** Receiver Operating Characteristic

List of Figures

2.1 Educational Data Mining Process	7
2.2 Applications of Educational Data Mining	8
2.3 Distribution of Papers that used different attributes for PSP problem	19
2.4 Student Performance Data Set	21
2.5 Educational Process Mining (EPM): A Learning Analytics Data Set ...	23

List of Tables

2.1	Student Performance Data Set Description	21
2.2	Students' Academic Performance Dataset Description	23

Chapter 1 Introduction

1.1 Overview

Arabic is one of the most widely spoken languages in the world, with millions of speakers distributed across many countries and regions. While Modern Standard Arabic is used as a common written language, people communicate on a day-to-day basis using various dialects that are very different in terms of vocabulary, pronunciation, and expressions. These dialectal variations are a challenge for Natural Language Processing (NLP) applications such as sentiment analysis, machine translation, chatbots, and speech recognition systems. Automatic dialect identification has become an important research area as accurately identifying the dialect of a text can significantly improve the performance of language technologies. Recent advances in Artificial Intelligence (AI), especially Deep Learning models such as BERT-based architectures, have shown promising results in dialect classification tasks. However, these models are typically black boxes, which makes it difficult to understand the rationale behind their predictions. In this context, Explainable Artificial Intelligence (XAI) techniques can be used to explain and interpret the model, allowing researchers and users to understand which linguistic features are relevant for classification decisions. The combination of AI and XAI, therefore, not only has high performance but also improves the credibility of dialect identification systems.

1.2 Problem Definition

Arabic dialects are very different from each other across countries and regions, which makes the automatic identification task a complex problem. Traditional language processing approaches often struggle to distinguish between similar dialects due to the common vocabulary and linguistic features. Moreover, several deep learning models provide accurate results, but they lack interpretability, making it difficult for the models to explain their predictions. Therefore, an intelligent and explainable system is required that can accurately identify Arabic dialects and countries and also provide interpretable explanations for its predictions.

1.3 Thesis Objectives

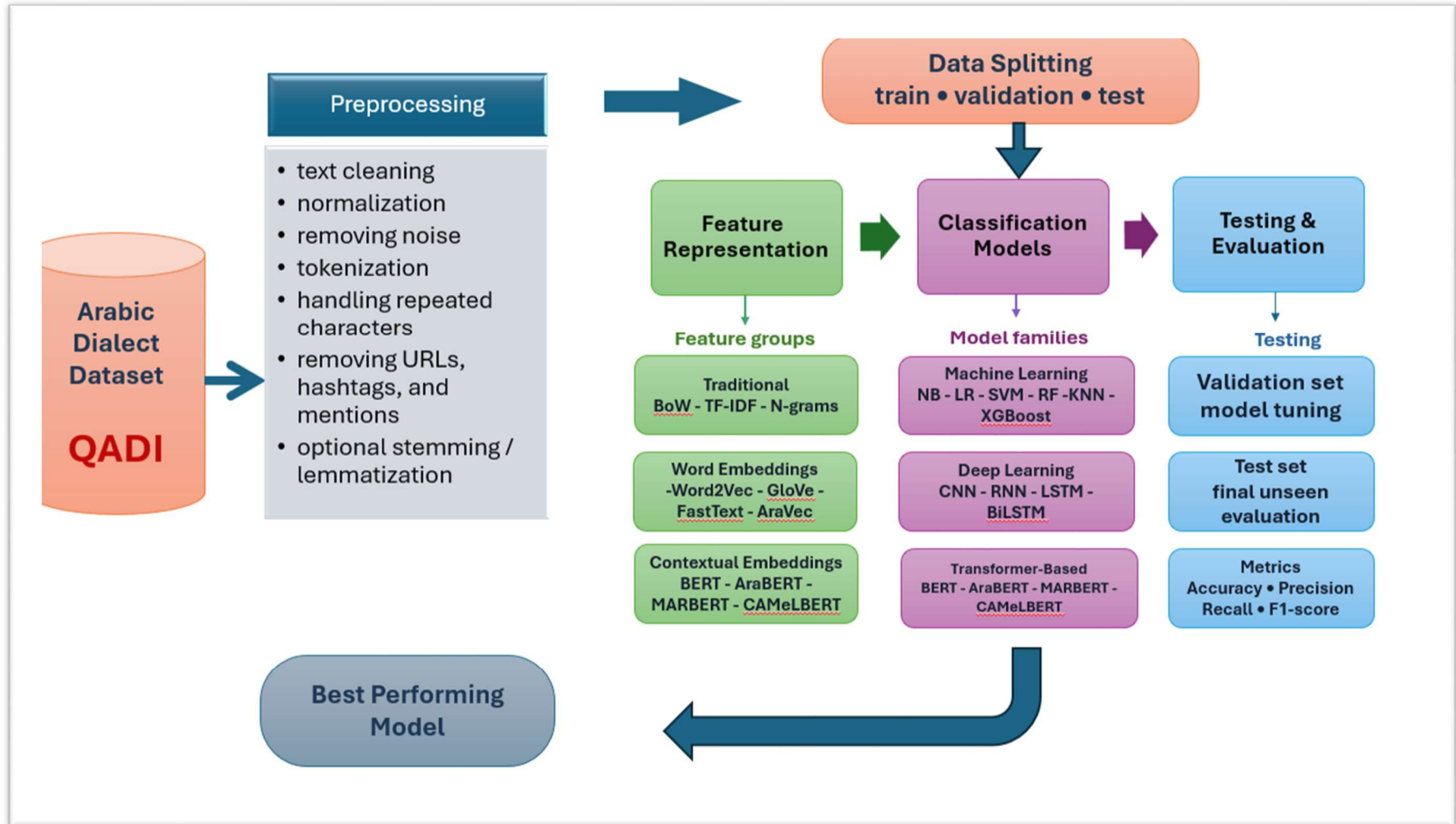
Develop an AI-based system for Arabic dialects and country identification from textual data, with an explanation of the prediction using Explainable AI techniques.

Specific Objectives:

- Collect and analyze large scale Arabic dialect dataset.
- Preprocess social media text data for use in machine learning and deep learning.
- Build traditional machine learning algorithms like Logistic Regression, Naïve Bayes and Support Vector Machine (SVM).
- Use transformer-based deep learning models, AraBERT and MARBERT.
- Classify Arabic texts to country level dialects of 18 Arab countries.
- Arabic dialects are classified into the following major regional groups: Gulf, Levantine, Egyptian, Maghrebi, Iraqi, Yemeni, and Sudanese.
- The developed models are evaluated using Accuracy, Precision, Recall and F1-score.
- Apply Explainable AI (XAI) methods such as SHAP and LIME to understand and explain the model predictions.
- Comparison of performance of traditional ML algorithms and transformer based models.
- Propose a framework for Arabic dialect identification for future Arabic Natural Language Processing (NLP) applications.

1.4 Thesis Contribution

1.5 Thesis Organization



Chapter 2 Background and Related Work

1 Background

Arabic is also a widely used language. It has flexible grammar and many regional dialects. Modern Standard Arabic (MSA) is used in formal communication and differs in vocabulary, pronunciation, grammar, and writing style. These variations make Arabic one of the most difficult languages for NLP applications.

Arabic Dialect Identification (ADI) is the problem of classifying Arabic text or speech into its dialect. ADI has been an important preprocessing step for many NLP applications such as machine translation, sentiment analysis, speech recognition, information retrieval, and conversational systems, and the correct dialect identification can significantly boost the overall system's performance. Despite the significant progress made in Arabic NLP, ADI is still a challenging problem due to the high linguistic similarity among dialects, spelling variations, code-switching, and the lack of standard writing conventions. These issues have led to the development of more advanced computational techniques, starting from the classical ML techniques to the more recent DL techniques and more recently, the Transformer-based techniques. The next sections present the main techniques for pre-processing, feature representation and computational frameworks used for Arabic Dialect Identification.

1.2 Natural Language Processing (NLP)

- **Data preprocessing:** is an essential step in preparing textual data for machine learning and Natural Language Processing tasks. It converts unstructured text into a structured representation while improving data quality by minimizing noise and inconsistencies. This process generally starts with data cleaning, where irrelevant components, including URLs, HTML tags,

emojis, special characters, extra whitespace, and other non-textual elements, are removed to produce cleaner and more reliable input for classification models [1].

- **Tokenization** is the process of converting text into a set of discrete linguistic units, known as tokens, such as words, sub-syllables, or sentences, making them amenable to processing and analysis by natural language processing algorithms, thus facilitating various tasks such as text classification and analysis [2].
- **Stop word removal** is a process that aims to eliminate common words that frequently appear in texts but do not add significant semantic value to content comprehension or text differentiation. This process is a preprocessing step in text processing, contributing to data reduction and improving the efficiency and accuracy of natural language processing models and information retrieval systems [2].
- **Stemming** is a text preprocessing technique that reduces inflected or derived words to their root or stem form by removing prefixes and suffixes. Its primary purpose is to group words with the same underlying meaning into a common representation, thereby reducing vocabulary size and improving the efficiency of information retrieval and Natural Language Processing (NLP) systems [2]

Stemming

Changing ————┘
Changed —————▶ Chang
Changes ————┘
Change ————┘

1.3 Text Representation

is the process of converting textual data into numerical representations that can be processed by machine learning and deep learning algorithms. These representations capture linguistic or semantic information from text and enable computational models to perform Natural Language Processing (NLP) tasks such as text classification, sentiment analysis, and machine translation [2] .

1.3.1 Bag of Words (BoW)

is a widely used text representation technique that represents a document based on the frequency of its words while disregarding their order. Rather than considering the sequence of words, the model focuses solely on the occurrence of each term within the document. As a result, two documents containing the same words with identical frequencies are treated as equivalent, regardless of the arrangement of those words[2] .

Mathematical Representation

$$d = \{tf_{vd}\}_{v \in V}$$

Assumption: The ordering of words does not matter, only what occurred



Figure 2.1 BAG of word [1]

1.3.2 Term Frequency–Inverse Document Frequency (TF-IDF)

is a text-weighting technique that evaluates the importance of a term by combining its frequency within a specific document (Term Frequency, TF) with its rarity across the entire document collection (Inverse Document Frequency, IDF). This approach assigns greater weights to terms that appear frequently in a particular document but occur less frequently in the overall corpus, thereby enhancing the effectiveness of text representation for information retrieval and text classification. Mathematical Representation [1].

Mathematical Representation

*The **Term Frequency (TF)** of a term t in document d is defined as:*

$$TF(t, d) = f_{t,d}$$

*The **Inverse Document Frequency (IDF)** is defined as:*

$$IDF(t) = \log \left(\frac{N}{df_t} \right)$$

*Finally, the **TF-IDF** weight is computed as:*

$$TF - IDF(t, d) = TF(t, d) \times IDF(t)$$

1.3.3 N-grams

are a text representation technique that divides text into contiguous sequences of n consecutive words. For example, a sequence of two consecutive words is known as a bigram, while a sequence of three consecutive words is called a trigram. By preserving neighboring word sequences, N-grams provide contextual information that helps improve the performance of various Natural Language Processing (NLP) tasks [1].

The following equations are adapted from [4]:

Equation (1): Chain Rule

$$P(w_1^m) = \prod_{i=1}^m P(w_i | w_1^{i-1})$$

Equation (2): N-gram Approximation

$$P(w_i | w_1^{i-1}) \approx P(w_i | w_{i-n+1}^{i-1})$$

1.3.4 Word2Vec

is a word embedding technique introduced by Google in 2013 to efficiently learn dense vector representations of words from large text corpora. It captures semantic and syntactic relationships between words by mapping those with similar meanings to nearby vector representations. The Word2Vec framework consists of two main architectures: Continuous Bag-of-Words (CBOW), which predicts a target word from its surrounding context, and Skip-gram, which predicts the surrounding context words from a target word[5] .

Mathematical Representation

The following equations are reproduced from [6]:

Word2Vec is implemented using two main architectures: Continuous Bag-of-Words (CBOW) and Skip-Gram. CBOW predicts the target word from its surrounding context, whereas Skip-Gram predicts the surrounding context words from the target word.

Continuous Bag-of-Words

$$P(w_i | w_{j(|j-i| \leq l, j \neq i)}) = \text{Softmax} \left(\mathbf{M} \left(\sum_{|j-i| \leq l, j \neq i} \mathbf{w}_j \right) \right),$$

Skip-Gram differs from CBOW by predicting the surrounding words given a center word rather than predicting the center word from its context[2].

$$P(w_j | w_i) = \text{softmax}(\mathbf{M} \mathbf{w}_i) (|j - i| \leq l, j \neq i),$$

where $P(w_j | w_i)$ is the probability of context word w_j given w_i , and $\mathbf{M}$ is the weight matrix. The loss function is defined similar to CBOW as

$$\mathcal{L} = - \sum_i \sum_{j (|j-i| \leq l, j \neq i)} P(w_j | w_i).$$

1.3.5 Global Vectors Algorithm

is an unsupervised word embedding technique developed by Pennington et al. that learns dense vector representations of words by analyzing global word co-occurrence statistics across an entire text corpus. Unlike local context-based models, GloVe captures semantic relationships by leveraging word–word co-occurrence information, enabling words with similar meanings to have similar vector representations [4].

Mathematical Representation

Objective Function [14]

$$J = \sum_{i,j=1}^V f(X_{ij}) (\mathbf{w}_i^T \tilde{\mathbf{w}}_j + b_i + \tilde{b}_j - \log X_{ij})^2$$

Weighting Function

$$f(x) = \begin{cases} (x/x_{\max})^\alpha, & \text{if } x < x_{\max}, \\ 1, & \text{otherwise.} \end{cases}$$

1.3.6 Fast Text

is a word embedding model built upon the Skip-gram architecture, where each word is represented as a collection of character n -grams rather than as a single unit. The vector representation of a word is obtained by summing the vector representations of its constituent character n -grams, allowing the model to capture morphological information and generate meaningful embeddings for rare and out-of-vocabulary words[3].

Mathematical Representation

The following equation is reproduced from [16]:

$$s(w, c) = \sum_{g \in G_w} \mathbf{z}_g^T \mathbf{v}_c$$

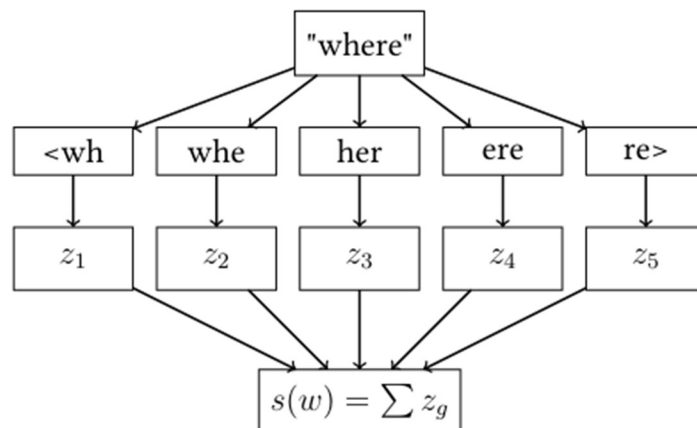


Figure 5: FastText word representation as the sum of its character n -gram vectors.

Figure 2. *FastText* word representation[3]

1.3.7 Long Short-Term Memory (LSTM)

is a type of Recurrent Neural Network designed to learn long-term dependencies in sequential data. It overcomes the vanishing gradient problem by using a memory cell and three gating mechanisms: the input gate, forget gate, and output gate. These gates regulate the flow of information, enabling the network to retain relevant information over long sequences while discarding irrelevant data. LSTM is widely used in natural language processing, speech recognition, machine translation, sentiment analysis, and time-series forecasting.

The LSTM model uses three gates to control the information stored, updated, and released by the memory cell:

$$\begin{aligned}f_t &= \sigma(U_f x_t + W_f h_{t-1} + b_f) \\i_t &= \sigma(U_i x_t + W_i h_{t-1} + b_i) \\c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(U_c x_t + W_c h_{t-1} + b_c) \\o_t &= \sigma(U_o x_t + W_o h_{t-1} + b_o) \\h_t &= o_t \odot \tanh(c_t)\end{aligned}$$

1.3.8 Gated Recurrent Unit (GRU)

is a type of recurrent neural network architecture introduced to efficiently model sequential data while reducing the computational complexity of traditional Long Short-Term Memory networks. GRU employs two gating mechanisms, the update gate and the reset gate, to regulate the flow of information through the network. This design enables the model to capture long-term dependencies, mitigate the vanishing gradient problem, and achieve competitive performance with fewer parameters and faster training. GRU is widely applied in natural language processing, speech recognition, machine translation, sentiment analysis, and time-series forecasting.

1.4 Evaluation Metrics

Accuracy is a performance evaluation metric used in machine learning and classification tasks to measure the proportion of correctly predicted instances among all observations. It is calculated by dividing the total number of correct predictions, including both true positives and true negatives, by the total number of samples.

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

2 Related Work

[1] Abdelminaam et al. proposed a framework for Arabic Opinion Mining (AOM) by comparing traditional Machine Learning (ML) and Deep Learning (DL) techniques over multiple Arabic dialects and Modern Standard Arabic (MSA). The authors utilized three publicly available Twitter datasets: Tweets Emoji Arabic Dataset (TEAD) with 5,615,943 tweets, Arabic Tweets Sentiment Analysis Dataset (ATSAD) with 58,751 tweets, and Arabic Sentiment Tweets Dataset (ASTD) with 10,006 tweets. These datasets have been collected from the Twitter platform and contain MSA and several Arabic dialects such as Egyptian, Gulf, Levantine, Maghrebi, Saudi, and Syrian dialects. In the preprocessing stage, Information Science Research Institute Stemmer (ISRI Stemmer) was used to remove noise from the data. Term Frequency–Inverse Document Frequency (TF-IDF) with N-grams and Arabic Vector (AraVec) word embeddings were used as feature representation methods for ML and DL models, respectively. The study evaluated six ML algorithms, namely Decision Tree (DT), Logistic Regression (LR), k-Nearest

Neighbors (KNN), Random Forest (RF), Support Vector Machine (SVM), and Naïve Bayes (NB), along with three DL models, namely Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Multiplicative Long Short-Term Memory (mLSTM). Hold-out and 10-fold cross-validation were used to evaluate the models on three benchmark datasets (TEAD, ATSAD, and ASTD). The experimental results showed that the mLSTM model performed the best, with 99.75% on the TEAD dataset, 97.52% on the ATSAD dataset and 96.45% on the ASTD dataset, outperforming all the traditional ML methods and the other DL models.

[2] Brihi et al. proposed a hybrid framework for Arabic Sentiment Analysis (ASA) targeting the Yemeni Arabic Dialect (YAD) by using ML and DL. dataset consisting of 45,862 manually annotated Facebook comments collected from the official Facebook pages of four Yemeni telecommunication companies: Yemen Mobile, Sabafon, YOU, and Yemen Telecom. In the preprocessing stage, they used a modified Information Science Research Institute (ISRI) Stemmer for the Yemeni dialect. They proposed a framework that evaluated lexicon-based methods, multiple ML classifiers, several DL models including Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), as well as hybrid architectures such as CNN-LSTM, CNN-BiLSTM, CNN-GRU, CNN-BiGRU, and stacked DL-ML models with Random Forest (RF) and Linear Support Vector Classifier (LinearSVC) as meta-learners. Experimental results proposed a hybrid stacking model that achieved the highest performance, reaching 94.71% accuracy using RF as the meta-learner and 94.28% accuracy using LinearSVC, outperforming all standalone lexicon-based, ML, and DL approaches for Yemeni Arabic sentiment analysis.

[3] Al-Anzi et al. This paper proposed a Region-Wise Arabic Dialect Identification (ADI) framework that classifies Arabic speech into regional dialects by utilizing a pre-trained speech classification model and deep learning techniques. They used a custom dataset consisting of 4,071 Arabic audio clips collected from television and radio news broadcasts for the preprocessing stage; they applied Discrete Fourier

Transform (DFT) followed by Mel-Frequency Cepstral Coefficients (MFCC) feature extraction. Feature selection was performed using Analysis of Variance (ANOVA), Kendall's Tau, Spearman's Correlation, and Recursive Feature Elimination (RFE), while data imbalance was handled with Adaptive Synthetic Sampling (ADASYN), Random Oversampling, and Synthetic Minority Oversampling Technique (SMOTE). The study employed the KLaam pre-trained speech classification model as the base model and tested Artificial Neural Network (ANN), One-Dimensional Convolutional Neural Network (1D CNN), and Two-Dimensional Convolutional Neural Network (2D CNN) architectures. The results obtained higher performance with 1D CNN reaching 94.28% testing accuracy and 95.55% validation accuracy, outperforming traditional machine learning models and the transformer-based AraBERT model for Arabic dialect classification.

[4] Tibi et al proposed an adaptive (ADI) framework using a Hamilton Neural Network (HNN) to improve Arabic dialect classification from speech. They used two benchmark speech datasets, namely the Linguistic Data Consortium (LDC) corpus and the (ADI) dataset, covering eight Arabic dialects (Tunisian, Algerian, Moroccan, Libyan, Egyptian, Gulf, Iraqi, and Levantine) collected from telephone conversations and Al Jazeera TV broadcasts. In the preprocessing stage, they applied Multi-Scale Product Analysis (MPA) to the speech signal to extract Glottal Closure Instant (GCI), Glottal Opening Instant (GOI), Open Quotient (Oq), Fundamental Frequency (F0), and Mel-Frequency Cepstral Coefficients (MFCCs) with their first-and second-order derivatives. The extracted features are used to train the proposed HNN classifier. The proposed HNN model was compared with the RNN and the CNN models. Experimental results showed that the proposed approach achieved 91.34% accuracy, 90.72% precision, 88.27% recall, and 87.44% F1-score, outperforming several state-of-the-art deep learning models for Arabic dialect identification.

[5] Abdelali, Ahmed, et al. proposed an (ADI) framework to identify 18 country-level Arabic dialects based on a newly constructed large-scale Twitter dataset, namely QCRI Arabic Dialect Identification

(QADI). QADI consists of about 540,000 tweets collected from 2,525 Twitter users in 18 Arab countries, with an average of 30,000 tweets per country. They automatically gathered Arabic tweets from Twitter and identified users based on the self-declared country information in their profiles during the preprocessing phase. They used a fastText classifier to filter MSA tweets, removed offensive content, split hashtags, normalized user mentions, URLs, and numbers, and kept only dialectal tweets. They evaluated SVM using character and word n-grams, Mazajak static embeddings, Multilingual BERT (mBERT), and AraBERT. The experimental results showed that AraBERT performed best with a macro-average F1-score of 60.6%, outperforming the performance of mBERT, SVM, and Mazajak embeddings for country-level Arabic dialect identification.

[6] Amjad Alsuwaylimi proposed a two-hybrid model that includes a BiLSTM with CAMeL BERT and a second model that combines the BiLSTM model with ALBERT. The ML models were Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbors (KNN), Naïve Bayes (NB), Extreme Gradient Boosting (XGBoost), and Stochastic Gradient Descent (SGD). Different word representation techniques were applied: TF-IDF, GloVe, and Word2Vec, and CAMeLBERT and ALBERT as transformer-based models. 198,233 comments were collected covering four major Arabic dialects: Egyptian, Jordanian, Gulf, and Yemeni. Regular expressions and the `ar_corrector` package in Python were Applied data cleaning and preprocessing to enhance the quality of data and accuracy of the models, such as the removal of duplicate characters, English characters, special symbols, and redundant spaces. The results showed the superiority of the CAMeLBERT + BiLSTM model with the highest accuracy of 87.67%, while the ALBERT + BiLSTM model achieved 86.51%, outperforming all the traditional and deep learning models used for comparison.

[7] Tahani et al. proposed a character-level model for Arabic dialect identification (ADI). Using deep learning and ML, the goal was to distinguish between four main Saudi dialects (Hijazi, Najdi, Southern, and Hasawi) in short social media texts. A corpus of 3,768 annotated sentences was collected, preprocessed, and manually labeled. (CNN) was used for character-level feature extraction, followed by

a Bidirectional Long Short-Term Memory (BiLSTM) network to capture sequential dependencies in the text. The results showed that machine learning outperformed deep learning, achieving 40.9% accuracy for four-class, 47% for three-class, and 64.5% for two-class classification.

[8] Brahim et al. proposed a model to automatically predict the Algiers dialect based on NLP techniques, machine learning algorithms (Support Vector Machine (SVM, Logistic Regression (LR, and MNB), and deep learning such as CNN and LSTM. The dataset used was the PADIC (Parallel Arabic Dialect Corpus) database, which contains parallel texts in Modern Standard Arabic and five Arabic dialects: the Algerian dialect, the Annaba dialect, the Tunisian dialect, the Palestinian dialect, and the Syrian dialect. The LSTM model had the best performance with an average of 94%, which fell to 97% when trained between Algerian and Standard Arabic.

[9] Hager et al. proposed a hybrid model to improve the accuracy of Arabic dialect identification and solve the challenges of the linguistic diversity among Arabic dialects. The model was built by merging Bert-Base-Arabertv02 and Dialectal-Arabic-XLM-R-Base converter models. The study used two datasets for Arabic dialect identification: the Shami dataset with 66,245 samples of four Levantine dialects (Jordanian, Lebanese, Palestinian, and Syrian) and the IADD dataset with 129,597 samples of five Arabic dialects (Levantine, Maghrebi, Gulf, Egyptian, and General Arabic). The preprocessing phase was performed, including tokenization, data cleaning, normalization, stop-word removal, and Arabic stemming. After that, 75% of the datasets were used for training and 25% for testing. The performance of the proposed model was evaluated against several other models, including CNN, LSTM, and GRU, and also the individual Bert-Base-Arabertv02 and Dialectal-Arabic-XLM-R-Base models. The results revealed that the proposed hybrid model outperformed all compared models with an accuracy of 89.73% on the Shami dataset and 93.06% on the IADD dataset.

[10] Wael et al. proposed a hybrid model to improve Arabic dialect detection on social media by integrating machine learning and deep learning techniques, including LSTM, BiLSTM, and Logistic Regression. The dataset used was a new dataset collected from Twitter between July 2022 and July 2023, consisting of 34,905 tweets in four Arabic dialects: Egyptian, Gulf, Jordanian, and Yemeni. The preprocessing removed special characters, English words and characters, English and Arabic numbers, and empty tweets, followed by Arabic stop-word removal. Three-word representation methods were then used: TF-IDF, Word2Vec, and GloVe. The results showed that the proposed model achieved the best performance with a high accuracy of 88.73%.

[11] Talafha et al. proposed an automatic Arabic dialect identification system based on contextual language representations learned by BERT. The proposed system employed Multi-Dialect Arabic BERT (MDABERT). The experiments were conducted on the NADI 2021 shared task dataset, which contains more than 21,000 labeled Arabic tweets collected from 21 Arab countries, representing informal dialectal text with country-level labels. Before training, the data were preprocessed by removing URLs, mentions, hashtags, emojis, and punctuation, followed by Arabic text normalization and tokenization using the BERT tokenizer. The model represented the text using contextual embeddings generated by MDABERT. The results showed that it achieved a Micro F1-score of 26.78%, outperforming traditional machine learning.

[12] El Mekki et al. proposed a transformer-based Multi-Task Learning (MTL) framework for Arabic dialect identification to improve dialect classification by jointly learning multiple related tasks through a shared BERT encoder with task-specific classification heads. The model was evaluated using the NADI 2021 dataset, which consists of Arabic tweets representing dialects from multiple Arab countries with country-level annotations. During preprocessing, the text was cleaned and normalized by removing URLs, mentions, repeated characters, and other unnecessary symbols before tokenization using the BERT tokenizer. The framework represented the text through deep contextual embeddings extracted from the shared BERT encoder, allowing knowledge to be shared across tasks and improving feature learning. Experimental results

demonstrated that the proposed model achieved higher F1-scores than the single-task baseline and exhibited better robustness and generalization across different Arabic dialects.

[13] Abdul-Mageed et al. proposed MARBERT, a transformer-based language model specifically designed for dialectal Arabic. The model was pre-trained on a corpus containing more than one billion Arabic tweets collected from Twitter, covering numerous regional dialects and informal writing styles. Before training, the data underwent preprocessing that included Arabic text normalization, removal of noisy characters, text cleaning, and WordPiece tokenization. Through large-scale pre-training, MARBERT learned rich contextual transformer embeddings capable of capturing dialectal variations effectively. The model was evaluated on several downstream natural language processing tasks, including Arabic dialect identification, where it achieved state-of-the-art performance and demonstrated that large-scale pre-training on dialectal Arabic significantly improves model performance.

[14] Inoue et al. proposed a task on Arabic dialect identification to identify both Modern Standard Arabic (MSA) and dialectal Arabic (DA) at the country and province levels. The authors proposed a Twitter-based dataset consisting of tweets collected from 21 Arab countries and 100 provinces, with 21,000 training, 5,000 development, and 5,000 test instances for each MSA and DA dataset, in addition to 10 million unlabeled tweets. During the preprocessing of the data, usernames, hyperlinks, and retweets were removed, whitespace was inserted between emojis, tweets with fewer than three Arabic tokens were discarded, non-Arabic tweets were filtered out using a language identification tool, and the data were separated into MSA and DA using an automatic dialect classification model. The top systems mainly utilized Transformer-based models, especially MARBERT, while some teams also used AraBERT and AraELECTRA. The best system, CairoSquad, achieved 32.26% macro-F1 (51.66% accuracy) for the country-level dialect identification, outperforming all systems that participated in the shared task.

[15] Antoun et al. proposed a transformer-based language model for Arabic language understanding, and it was evaluated on several NLP tasks, such as sentiment analysis, named entity recognition (NER), and

question answering (QA). The authors used a large-scale Arabic corpus, consisting of around 70 million sentences (24 GB) collected from Arabic Wikipedia, news websites, the 1.5 billion Arabic Corpus, and the OSIAN Arabic News Corpus. During the preprocessing phase, duplicate sentences were removed; Arabic text was segmented into prefixes, stems, and suffixes using Farasa, followed by Sentence Piece tokenization to generate sub word units and reduce the size of the vocabulary. The proposed AraBERT model is based on the BERT-Base architecture and trained using Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The experimental results showed that AraBERT outperforms multilingual BERT and previous state-of-the-art models, achieving 96.2% accuracy on the HARD dataset, 92.6% on ASTD, 93.8% on AJGT, and the highest macro F1-score of 84.2% for NER and 62.7% for question answering. This makes AraBERT the best-performing model across most evaluated Arabic NLP tasks.

	Models	Dataset	Link	Limitation	Base Model	Result
[1] Abdelminaam et	ML: DT, LR, KNN, RF, SVM, NB. DL: LSTM, GRU, mLSTM (proposed)	TEAD, ATSAD, ASTD	IEEE Access, 2021. DOI: 10.1109 /ACCE SS.2021 .309417 3	Performance decreases on imbalanced datasets (especially ASTD). The study is limited to recurrent models (LSTM, GRU, mLSTM) and does not investigate transformer-based architectures such as BERT or AraBERT.	AraVec (Word2Vec pretrained embeddings)	The proposed mLSTM achieved the best performance with 99.75% on TEAD, 97.52% on ATSAD, and 96.45% on ASTD, outperforming all ML models and the other DL models.
[2] Brihi et al.	Lexicon-Based, ML (LR, SVM, LinearSVC, SGD, RF, AdaBoost, XGBoost, MNB, BNB, KNN), DL (MLP, CNN, RNN, LSTM, GRU), Hybrid (CNN-LSTM, CNN-	45,862 manually annotated Facebook comments in the Yemeni Arabic Dialect (YAD) collected from Yemen Telecom, Yemen Mobile, YOU, and Sabafon	IEEE Access (2025). DOI: 10.1109 /ACCE SS.2025 .359320 4	The study focuses only on the Yemeni dialect and one application domain (telecommunication Facebook comments). The authors also recommend using transformer-based models (e.g., BERT/AraBERT) in future work to capture deeper contextual information.	Word2Vec (CBOW + Skip-gram) with TF-IDF, Lexicon features, and Feature Stacking	The proposed Hybrid DL-ML Stacking model achieved the best performance with 94.71% accuracy using Random Forest (RF) as the meta-learner and 94.28% accuracy using LinearSVC, outperforming standalone Lexicon-Based, ML, and DL models.

	BiLSTM, CNN-GRU, CNN- BiGRU, Stacked DL- ML)					
[3] Al-Anzi et al.	ANN, 1D CNN, 2D CNN with feature selection (ANOVA, Kendall's Tau, Spearman's Correlation, RFE) and data augmentation (ADASYN, Random	4,071 Arabic audio clips covering MSA, Egyptian, Gulf, Levantine, and North African dialects	https://www.mdpi.com/2076-3417/15/12/6516	The study focuses only on speech-based dialect identification rather than text. The dataset covers five major dialects, while several low-resource dialects (e.g., Mauritanian, Bahraini, Sudanese) are not included. The authors recommend expanding the dataset and dialect coverage in future work.	Klaam Pre-trained Speech Classification Model	The proposed framework achieved 94.28% testing accuracy and 95.55% validation accuracy using the 1D CNN model, outperforming traditional ML models and AraBERT for Arabic dialect classification.

	Oversampling, SMOTE)					
[4] Tibi et al	Hamilton Neural Network (HNN) (Proposed), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN)	LDC (Linguistic Data Consortium) and ADI (Arabic Dialect Identification) datasets, covering 8 Arabic dialects (Tunisian, Algerian, Moroccan, Libyan, Egyptian, Gulf, Iraqi, and Levantine). Speech collected from telephone	https://beej.org/index.php/EEI/article/view/8165/4121	The model is evaluated only on speech-based dialect identification and depends on handcrafted acoustic/prosodic features (GCI, GOI, Oq, F0, MFCC). The authors recommend expanding the dataset with more speaker variations and developing additional multi-view features to improve performance.	Hamilton Neural Network (HNN) with Multi-Scale Product Analysis (MPA) feature extraction	The model achieved 91.34% Accuracy, 90.72% Precision, 88.27% Recall, and 87.44% F1-score, outperforming RNN, CNN, and several recent state-of-the-art dialect identification models.

		conversations and Al Jazeera TV broadcasts.				
[5] Abdelali, Ahmed, et al.	SVM, mBERT, AraBERT, Mazajak Embeddings	QADI dataset: 540,000 tweets from 2,525 Twitter users across 18 Arabic countries	arXiv:2005.06557	Limited to Twitter data and 18 country-level dialects. Four Arab countries (Mauritania, Somalia, Djibouti, and Comoros) were excluded because sufficient Twitter data were unavailable. Performance also decreases geographically similar dialects due to linguistic overlap.	AraBERT	AraBERT achieved the best performance with a Macro-average F1-score of 60.6%, outperforming mBERT, SVM, and Mazajak embeddings on the QADI dataset.

[6] Amjad Alsawaylimi	machine learning (SVM, LR, RF, DT, KNN)) deep learning BiLSTM	Data set created by the researcher himself	Arabic Dialect Dataset	The study is limited to four Arabic dialects.	CAMELBERT + BiLSTM	CAMELBERT + BiLSTM achieved an accuracy of 87.67%, outperforming all machine learning and deep learning models.
[7] Tahani et al.	machine learning models (Logistic Regression, SVM, Bayesian models, and SGD) and a deep learning approach combining CNN and BiLSTM	3,768 short sentences written in the Saudi dialect on social media	The data presented in this study are available upon request from the author	The great similarity between the dialects made the process of distinguishing between them difficult and reduced the accuracy of the models.	Logistic Regression, SVM, Bayesian models, and SGD	accuracies of 40.9% for four-class classification, 47% for three-class, and 64.5% for two-class tasks

[8] Brahim et al.	Machine Learning models) SVM, LR, MNB (Deep Learning models) CNN, (LSTM (	PADIC (Parallel Arabic Dialect Corpus)	PADIC download SourceForge.net	It relies on the PADIC database, which includes a limited number of Arabic dialects.	The LSTM and CNN models achieved the best performance	Deep learning models outperform traditional models in dialect recognition
[9] Hager et al.	Stacking-Transformer (Bert-Base-Arabertv02 + Dialectal-Arabic-XLM-R-Base + Random Forest Meta-Learner	Shami (66,245 samples) and IADD (129,597 samples)	https://shorturl.cc/1tfqV And https://shorturl.cc/1yMpg	data imbalance, high computational cost, and challenges in generalizing to underrepresented Arabic dialect	Bert-Base-Arabertv02 and Dialectal-Arabic-XLM-R-Base.	achieving 89.73% accuracy on Shami and 93.06% accuracy on IADD

[10] Wael et al.	Hybrid Deep Learning Model (LSTM + BiLSTM + Logistic Regression with Attention Mechanism)	A newly collected Twitter dataset consisting of 34,905 manually annotated tweets	Arabic dialect	The study was limited to four Arabic dialects and relied solely on data from the Twitter platform.	LSTM + BiLSTM + Logistic Regression (Hybrid Model with Attention Mechanism)	Achieved 88.73% accuracy, outperforming all baseline models.
Talafha et al. (2021)[11]	Multi-Dialect Arabic BERT, Ensemble	NADI 2021	https://aclanthology.org/2021.vardial-1.38/	Country-level difficulty	Ensemble Arabic BERT	Micro F1=26.78%
El Mekki et al. (2022)[12]	BERT+MTL	NADI 2021	https://aclanthology.org/2022.wanlp-1.50/	High computational cost	BERT+MTL	Higher F1 than single-task

Abdul-Mageed et al. (2021)[13]	MARBERT	1B Arabic Tweets	https://aclanthology.org/2021.acl-long.551/	Large model	MARBERT	State-of-the-art
[14] Inoue et al.	MARBERT, AraBERT, AraELECTRA, mBERT, Transformer-based models, Classical ML, Neural Networks	NADI 2021 Dataset collected from Twitter , containing MSA and Dialectal Arabic (DA) tweets from 21 Arab countries and 100 provinces . The labeled dataset includes 21,000 training , 5,000 development , and 5,000 test tweets for each of the MSA and DA datasets. An additional 10	https://aclanthology.org/2021.vardial-1/	The task is highly challenging due to the fine-grained distinction between dialects at the country and province levels. Labels were assigned based on users' geographical locations rather than manual linguistic annotation, which may introduce labeling noise.	MARBERT was the strongest base Transformer model used by the top-performing systems (especially CairoSquad and CS-UM6P). Other teams also employed AraBERT , AraELEC	The CairoSquad system achieved the best overall performance, obtaining 22.38% Macro-F1 (35.72% Accuracy) for Country-level MSA, 32.26% Macro-F1 (51.66% Accuracy) for Country-level Dialect Identification, 6.43% Macro-F1 (6.66% Accuracy) for Province-level MSA, and 8.60% Macro-F1 (9.46% Accuracy) for Province-level Dialect Identification,

		million unlabeled Arabic tweets were provided for optional semi-supervised learning.			TRA, and mBERT.	outperforming all participating systems.
[15] Antoun et al.	AraBERT (v0.1 & v1)	Pre-training: Arabic Wikipedia, Arabic News Websites, 1.5 Billion Arabic Corpus, OSIAN Arabic News Corpus (≈ 70M sentences, 24GB). Evaluation: HARD, ASTD, ArSenTD-Lev, AJGT, LABR, ANERcorp, ARCD.	https://aclanthology.org/2020.osact-1.2/	The model was mainly pre-trained on Modern Standard Arabic (MSA) , which may limit its understanding of some Arabic dialects. In addition, the first version required Farasa segmentation as an external preprocessing tool.	BERT-Base Transformer (12 layers, 768 hidden size, 12 attention heads, 110M parameters)	Best performance: HARD (96.2% Accuracy), ASTD (92.6% Accuracy), AJGT (93.8% Accuracy), NER (ANERcorp: 84.2% Macro F1), QA (ARCD: 62.7% Macro F1), outperforming multilingual BERT and previous state-of-the-art models.

Chapter 4 Proposed Approach (Methodology)

Chapter 5: Experimental Results and Discussion

Chapter 6 Conclusion and Future Work

References