People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Ferhat Abbas Setif -1-
Faculty of Sciences
Department of Computer Science

Towards a Robust Natural Language Understanding: Bridging the Low-Resource Gap in Algerian Dialect through MSA Knowledge Transfer

Ph.D. Thesis by: MAFAZA CHABANE

This thesis satisfies the requirements for the degree of Ph.D. in Computer Science - Machine Learning and Intelligent Systems

Committee of examiners

Houceme Mansouri	Prof.	Univ. Ferhat Abbes Setif 1	President
Fouzi Harrag	Prof.	Univ. Ferhat Abbes Setif 1	Supervisor
Khaled Shaalan	Prof.	British University in Dubai, UAE	Co-Supervisor
Salim Chikhi	Prof.	Univ. Abdelhamid Mehri Constantine	Examiner
Abdelhamid Djeffal	Prof.	Univ. Mohamed Khider Biskra	Examiner
Yacine Slimani	MCA.	Univ. Ferhat Abbes Setif 1	Examiner

PUBLICLY DEFENDED ON: DECEMBER 07, 2025

Abstract

Natural Language Processing (NLP) technologies have seen remarkable progress in recent years, unlocking new possibilities across domains such as education, healthcare, and social media. However, this progress remains largely confined to high-resource languages, leaving low-resource varieties, particularly Arabic dialects like Algerian Arabic underrepresented. These dialects face compounded challenges: lack of standardized orthography, code-switching with French and Modern Standard Arabic (MSA), rich morphological structures, and a persistent scarcity of annotated data. This thesis addresses these limitations by leveraging cross-lingual transfer learning and multitask learning, techniques to bridge the resource divide between MSA and Algerian Arabic. Central to this research is the hypothesis that linguistic proximity between MSA and Algerian Arabic can be systematically exploited to enhance NLP model performance in dialectal tasks. To validate this, a series of experiments was conducted, beginning with evaluations of classical machine learning models and pre-trained transformer architectures, revealing their limitations when applied to unstructured dialectal data. These observations motivated the development of two novel computational frameworks tailored for low-resource scenarios. The first contribution, WASL-DI, is a hybrid dialect identification system that combines contextual embeddings from the CAMeLBERT MSA model with semantic representations derived from FastText. This dual-path architecture captures both deep contextual and subword-level features, making it robust against noise and lexical variation common in informal dialectal content. It achieved a peak accuracy of 99.24% on the dataset used, outperforming benchmark models such as DziriBERT and MDA-BERT. The second major innovation is SILAA-SA, a multitask learning framework for sentiment analysis. It incorporates a Mixture of Experts (MoE) mechanism to dynamically share knowledge between MSA and dialectal inputs. The model uses shared layers for general language understanding and task-specific experts for capturing dialectal nuances, ensuring efficient knowledge transfer without semantic interference. Extensive experiments across multiple dialectal sentiment datasets show that SILAA-SA outperforms traditional singletask models and adapts well to cross-domain tasks such as fake news detection. SILAA-SA achieved 86.81% accuracy on the FASSILA dataset and outperformed existing models across several dialectal benchmarks, including MAC, MYC, TSAC, and ArSarcasm-v2, with strong cross-domain performance on fake news detection tasks as well. Ablation studies further confirmed the effectiveness of architectural components like embedding fusion and MoE design in achieving performance gains. Beyond model accuracy, this work addresses broader concerns in dialectal Natural Language Processing (NLP) by providing reproducible pipelines, annotated datasets, and adaptable architectures to support future research in Arabic and other low-resource languages. Importantly, the research also promotes inclusivity by extending NLP capabilities to communities often excluded from technological advances. By demonstrating how high-resource language assets like MSA can be repurposed for low-resource dialects, the thesis sets forth scalable and practical strategies that address real-world data limitations. In conclusion, this work

makes substantial contributions to the field of cross-lingual and multitask learning for dialectal Arabic. It lays a robust foundation for further exploration in areas such as equitable language technology. The outcomes have direct implications for building more inclusive AI systems capable of understanding the diverse linguistic landscape of the Arabic-speaking world.

RÉSUMÉ

Les technologies de traitement automatique du langage naturel (TALN) ont connu ces dernières années des avancées remarquables, ouvrant la voie à de nouvelles applications dans des domaines variés tels que l'éducation, la santé ou encore les médias sociaux. Toutefois, ces progrès demeurent principalement centrés sur les langues disposant de ressources abondantes, reléguant les variétés peu dotées, notamment les dialectes arabes comme l'arabe algérien à la marge des innovations linguistiques. Ces dialectes se heurtent à une série d'obstacles : absence d'orthographe normalisée, alternance linguistique fréquente avec le français et l'arabe standard moderne (ASM), complexité morphologique marquée, et rareté persistante des corpus annotés. La présente thèse s'attache à dépasser ces limites en s'appuyant sur des techniques d'apprentissage multitâche et de transfert cross-lingue pour réduire l'écart entre l'ASM et l'arabe algérien. Au cœur de cette recherche réside l'hypothèse que la proximité linguistique entre l'ASM et l'arabe algérien peut être exploitée méthodiquement afin d'améliorer les performances des modèles NLP dans des tâches dialectales. Pour tester cette hypothèse, une série d'expérimentations a été menée, en commençant par une évaluation de modèles classiques d'apprentissage automatique et d'architectures pré-entraînées de type transformeur. Ces premiers tests ont révélé les limites de ces approches lorsqu'elles sont confrontées à des données dialectales non structurées, motivant le développement de deux cadres computationnels innovants, conçus pour les contextes à faibles ressources. Le premier, WASL-DI, est un système hybride d'identification dialectale combinant les représentations contextuelles du modèle CAMeLBERT (ASM) et les vecteurs sémantiques issus de FastText. Cette architecture bi-canal capte à la fois la richesse contextuelle profonde et les motifs morphologiques de surface, renforçant sa robustesse face au bruit lexical et à la variation dialectale. Il a atteint une accuracy de 99.24% sur les données utilisées, surpassant des modèles de référence comme DziriBERT et MDA-BERT. Ce système a obtenu des résultats de pointe sur plusieurs benchmarks, notamment pour les variantes algériennes sous-représentées, et a fait l'objet d'une publication dans une revue à comité de lecture. La deuxième contribution majeure, SILAA-SA, repose sur une architecture multitâche dédiée à l'analyse des sentiments. Elle intègre un mécanisme de type Mixture of Experts permettant une distribution dynamique des connaissances entre l'ASM et les entrées dialectales. Les couches partagées garantissent une compréhension linguistique globale, tandis que les experts spécifiques capturent les nuances dialectales sans provoquer d'interférences sémantiques. Le modèle SILAA-SA a atteint une accuracy de 86.81% sur FASSILA, et a surpassé les modèles existants sur les corpus MAC, MYC, TSAC et ArSarcasm-v2, avec de solides performances en détection des fausses nouvelles dans des contextes inter-domaines. Les expériences menées sur plusieurs corpus dialectaux de sentiments démontrent que SILAA-SA surpasse les modèles classiques monolingues, avec une capacité notable de généralisation à des tâches croisées comme la détection des fausses nouvelles, ces deux systèmes ont été rigoureusement évalués. Les études d'ablation ont souligné l'impact déterminant des composants architecturaux, tels que la fusion des embeddings et le mécanisme Mélange d'experts (MdE). Au-delà de l'aspect technique, cette recherche vise à élargir les possibilités du TALN à des communautés souvent exclues des avancées technologiques. En définitive, cette thèse constitue une contribution significative à l'apprentissage croisé et multitâche pour les dialectes arabes. Elle pose les bases d'une extension des capacités du TALN à d'autres langues à faibles ressources, tout en œuvrant à un traitement linguistique plus équitable dans le monde arabe.

ملخص

شهدت تقنيات معالجة اللغة الطبيعية (NLP) في السنوات الأخيرة تطوّرا ملحوظًا، ما فتح آفاقًا جديدة في مجالات متعددة كالتعليم والرعاية الصحية ووسائل التواصل الاجتماعي. غير أن هذه الطفرة التكنولوجية بقيت محصورة في اللغات الغنية بالموارد، في حين ظلّت اللغات والأنماط اللغوية قليلة الموارد، مثل اللهجات العربية عموماً واللهجة الجزائرية على وجه الخصوص، تعاني من ضعف التمثيل في هذا المجال. وترجع هذه الفجوة إلى جملة من التحديات المتراكبة، منها غياب نظام كتابي موحد، وانتشار التداخل اللغوي مع الفرنسية والعربية الفصحى، والتراكيب الصرفية المعقدة، فضلاً عن ندرة البيانات المشروحة. وتسعى هذه الأطروحة إلى معالجة هذه العقبات من خلالتوظيف تقنيات التعلم متعدد المهام والتعلم الانتقالي بين اللغات لسد الفجوة بين العربية الفصحى واللهجة الجزائرية. وتنطلق هذه الدراسة من فرضية علمية مفادها أن القرب اللغوي بين العربية الفصحى واللهجة الجزائرية يمكن استغلاله بشكل منهجي لتعزيز أداء نماذج NLP في المهام الخاصة باللهجات. وللتحقق من هذه الفرضية، تم إجراء سلسلة من التجارب، بدأت بتقييم نماذج التعلم الآلي التقليدية وبعض بيانات المحولات المدربة مسبقًا، والتي كشفت عن محدودية هذه الأساليب عند التعامل مع البيانات اللهجية غير المهيكلة. وقد قادت هذه النتائج إلى تصميم إطارين حسابيين مبتكرين يلائمان سيناريوهات نقص الموارد. الإسهام الأول، WASL-DI، هو نظام هجيني لتحديد اللهجات، يجمع بين التمثيلات السياقية المشتقة من نموذج CAMeLBERT المدرب على العربية الفصحى، والتمثيلات الدلالية المستخرجة من FastText. وتعتمد هذه البنية ثنائية المسار على الدمج بين العمق السياقي والدقة المقطعية، ما يمنح النظام مرونة في التعامل مع الضجيج والتنوع المعجمي في المحتوى غير الرسمي. وقد حقق دقة رائدة بلغت 99.24% على مجموعة البيانات المستخدمة، متفوقًا على نماذج قوية مثل DziriBERT وMDA-BERT. أما الإسهام الثاني، AS-AALIS، فهو إطار تعلم متعدد المهام لتحليل المشاعر، يعتمد على آلية "مزيج الخبراء (Mixture of Experts)" لتيسير التبادل الديناميكي للمعرف بين العربية الفصحى واللهجة الجزائرية. وتُستخدم فيه طبقات مشتركة للفهم العام للغة، إلى جانب خبراء مخصصين لرصد الخصوصيات اللهجية، مما يضمن نقلًا فعالاً للمعرفة دون حدوث تداخل دلالي. وقد حقق النموذج دقة بلغت 86.81% على مجموعة بيانات FASSILA، متفوقًا على النماذج الحالية عبر عدة مجموعات لهجية، مع أداء قوي كذلك في مهام كشف الأخبار الزائفة عبر مجالات متعددة. وقد أظهرت التجارب الشاملة التي أجريت على عدة مجموعات بيانات للهجات أن هذا النموذج يتفوق على النماذج التقليدية أحادية المهمة، ويُظهر كفاءة عالية عند تطبيقه على مهام مجاورة مثل كشف الأخبار الزائفة، خضعت هذه الأطر لتقييم دقيق. وأظهرت دراسات الإقصاء (ablation) فعالية عناصرها البنيوية مثل دمج التمثيلات وآلية الخبراء. وبالإضافة إلى الجوانب التقنية، تسعى إلى تعزيز الشمول اللغوي من خلال توفير أدوات رقمية تخدم المجتمعات المهمشة لغويًا. ختامًا، تطرح هذه الدراسة إسهامات بارزة في مجال التعلم متعدد المهام والانتقالي للهجات العربية، وتضع أساسا قويًا لاستكشافات مستقبلية تشمل تطوير تقنيات لغوية عادلة. كما تسهم مباشرة في بناء أنظمة ذكاء اصطناعي أكثر شمولاً قادرة على فهم المشهد اللغوي المتنوع في العالم العربي.

ACKNOWLEDGEMENTS

I would like to thank Prof. Fouzi Harrag for his continuous guidance, and the considerable time and effort he dedicated throughout the course of this research. I also extend my appreciation to Prof. Khaled Shaalan for his guidance, support, and for kindly accepting to co-supervise my work.

I also extend my sincere appreciation to the members of the PhD defense committee for their time, valuable feedback, and evaluation of this work.

Dedication

وَمَا تَوْفِيقِيٓ إِلَّا بِٱللَّهِ عَلَيْهِ تَوَكَّلْتُ وَإِلَيْهِ أُنِيبُ

Thanks to Allah, the Most Gracious, the Most Merciful. He gave me the strength to see every setback as a test, and filled my heart with the certainty that none but Him holds my destiny, and that certainty is what kept me going. May this work serve as a humble reflection of His blessings and a reminder that, with trust in Him no path is beyond reach.

To my beloved husband, whose unwavering support, patience, and belief in me from the very beginning have been the bedrock of this achievement. He envisioned this journey for me even before I considered it myself, and encouraged my very first step. Through every high and low, he stood beside me, offering strength when I needed it most. Quietly, and without ever seeking recognition, he was there more than anyone. It is for him, and because of him, that I undertook and completed this journey. This thesis, born of his love and support, is as much his as it is mine.

To my cherished parents. To my beloved father, who taught me to pursue knowledge with integrity in a world that often overlooks it. To my dearest mother, whose sacrifices and ceaseless prayers have been my greatest shield, her love, and patience were my strength. My hard work may have played a part but I have no doubt that it was her duas that truly brought me here. May Allah make me their Sadaqah Jariyah, a source of ongoing reward for them in this life and the next.

To my wonderful brothers and sisters, and my dear nieces and nephews, for their unwavering support, endless encouragement, the warmth of their love, and the comfort their presence brings.

To myself, for the countless hours of dedication, perseverance, and, above all, sheer survival. Through the long nights, the self-doubt, the moments when giving up seemed easier, and the relentless pressure, I kept pushing forward. This thesis is a testament to resilience, to overcoming not only the natural challenges of academia but also those unnecessarily placed in my path. A battle won against the odds, a journey of endurance, and proof that perseverance prevails.

And to my cat, Max, for doing nothing but being his adorable, lazy self, providing endless cuddles. His presence was a constant reminder to take breaks, breathe, and appreciate life's simple joys.

Finally, to my Mac, which wrote the very first word of this journey and the very last, enduring countless sleepless nights running code, a silent yet indispensable companion in this journey.

TABLE OF CONTENTS

List of Tables

List of Figures

Figure **Page**

Acronyms

ADI Arabic Dialect Identification

AI Artificial Intelligence

BERT Biderctional Encoder Representations from Transformers

BiGRU Bidirectional Gated Recurrent Unit

BiLSTM Bidirectional Long Short-Term Memory

BoW Bag-of-Words

CA Classical Arabic

CBoW Continuous Bag-of-Words

CE Character-level Embeddings

CNB Complement Naïve Bayes

CNN Convolutional Neural Network

DA Dialectal Arabic

DL Deep Learning

DNNs Deep Neural Networks

GloVe Global Vectors for Word Representation

GPT Generative Pre-Trained Transformers

GRU Gated Recurrent Unit

KNN K-Nearest Neighbors

LinearSVC Linear Support Vector Classifier

LLMs Large Language Models

LSTM Long-Short Term Memory

MADAR Multi-Arabic Dialect Applications and Resources

ML Machine Learning

MLDID Multi-label Arabic Dialect Identification

MLM Masked Language Modeling

MLP Multi-Layer Perceptron

MNB Multinomial Naïve Bayes

MoE Mixture of Experts

MRMR Minimum Redundancy Maximum Relevance

MSA Modern Standard Arabic

NADI Nuanced Arabic Dialect Identification

NB Naïve Bayes

NLP Natural Language Processing

NSP Next Sentence Prediction

OOV Out Of Vocabulary

RAG Retrieval-Augmented Generation

ReLU Rectified Linear Unit

RF Random Forests

RNN Recurrent Neural Network

RU-S Random Under-Sampling

SGD Stochastic gradient descent

SMOTE Synthetic Minority Over-sampling Technique

SVM Support Vector Machine

SVO Subject-Verb-Object

SWE Static Word Embeddings

tanh Hyperbolic Tangent

TF-IDF Term Frequency-Inverse Document Frequency

VoC Voting Classifier

VSO Verb-Subject-Object

XGBoost Extreme Gradient Boosting

CHAPTER 1

INTRODUCTION

Over the past decade NLP has made significant strides, largely driven by the availability of large-scale datasets and the development of advanced pretrained models. These advancements have led to substantial improvements in a variety of NLP tasks, such as speech recognition, sentiment analysis, and machine translation. However, the benefits of these innovations have not been equally distributed. While high-resource languages have reaped the rewards of these developments, many other languages, the low-resourced ones, remain underserved due to a lack of data. This disparity highlights a pressing challenge in NLP: ensuring that technological progress is inclusive and accessible to all languages, regardless of their resource availability.

1.1 The Gap Between High-Resource and Low-Resource Languages

Language is a defining feature of humanity, serving as the medium through which individuals share ideas, express complex emotions, and transfer knowledge across generations [149]. This capacity for expression has enabled humanity to build societies and achieve cultural and scientific advancements that shape the human experience. With over 7,000 languages spoken globally [69], language diversity serves as a testament to the cultural richness of humanity. Each language carries unique features in its grammar, vocabulary, and phonology, reflecting the values and worldviews of its speakers. However, this diversity also creates challenges, especially in a rapidly globalizing world where cross-linguistic communication is increasingly vital for social, economic, and technological interactions [151].

In the realm of technology, language has taken on an equally critical role, becoming a

cornerstone of Artificial Intelligence (AI) development. In the digital age, language is not just a means of communication but also a critical interface between humans and machines. AI-powered applications such as machine translation, speech recognition, and chatbots rely on the ability to process and understand human language, making linguistic data an essential resource for training and improving these systems. Whether through text, speech, or other modalities, the ability of machines to process and understand language profoundly impacts areas like education [22], healthcare [92], and commerce [144]. However, the availability of such resources varies significantly across languages, creating a gap between high-resource and low-resource languages that has hindered the development and accessibility of AI technologies [100].

High-resource languages, such as English, Mandarin, and Spanish [106], benefit from an abundance of linguistic data, including large-scale text corpora, annotated datasets, and pre-trained language models. These resources are the result of decades of research, investment, and digitalization efforts, which have made it possible to collect and curate vast amounts of language data. For example, English-language datasets like the Common Crawl [65] and the OpenWebText corpus [83] contain billions of words, enabling the training of highly accurate NLP models [49]. Similarly, pre-trained models such as GPT [13] and BERT [67] have been pretrained on extensive datasets, allowing them to perform complex tasks like text generation, sentiment analysis, and question answering with remarkable precision. The availability of these resources has propelled high-resource languages to the forefront of AI innovation, ensuring that speakers of these languages have access to cutting-edge technologies.

In contrast, low-resource languages face significant challenges due to the scarcity of linguistic data and computational resources. These languages, which include many regional dialects and minority languages, often lack the annotated datasets and digital text records necessary to train effective NLP models. For instance, while MSA has a relatively well-established presence in digital form, many Arabic dialects such as Algerian Arabic, and Moroccan Arabic —are underrepresented in linguistic datasets. This scarcity is compounded by the informal nature of these dialects, which are primarily used in spoken communication and rarely appear in standardized written form [153]. As a result, developing NLP models for low-resource languages requires overcoming significant barriers related to data collection, annotation, and standardization.

The gap between high-resource and low-resource languages is further exacerbated by the linguistic diversity that characterizes many low-resource languages. while languages often benefit from standardized orthographies and well-documented grammatical rules, some low-resource languages particularly dialects or regional varieties frequently exhibit variations in syntax, morphology, and phonetics [99]. For example, Algerian Arabic incorporates influences from French, Berber, and Turkish, resulting in a rich but complex linguistic landscape [136]. These variations make it difficult to create universal NLP solutions that can accurately process and generate text across different dialects and contexts. Additionally, in cases where standardized orthography is lacking, the same word or phrase can be written in multiple ways, further

complicating the development of effective NLP models. However, it is important to note that not all low-resource languages face these challenges, many simply lack sufficient annotated data or computational tools despite having well-defined linguistic structures.

The disparity in language resources has far-reaching implications for the accessibility and inclusivity of AI technologies. Speakers of high-resource languages enjoy access to a wide range of AI-driven tools and services, from voice assistants to real-time translation systems, that enhance communication and productivity. In contrast, speakers of low-resource languages often find themselves excluded from these advancements, as the lack of linguistic data and computational resources hinders the development of effective NLP models [100]. This imbalance not only limits access to technology but also perpetuates inequities in global communication and innovation.

1.2 Motivation

The motivation for this research stems from the urgent need to bridge the resource gap between high-resource and low-resource languages, ensuring that technological advancements are accessible to all speakers. While languages like English have benefited from extensive NLP research, many regional dialects and minority languages remain underserved. Among Arabic languages, MSA serves as a high-resource language, providing a wealth of resources such as large-scale text corpora, annotated datasets, and pre-trained models. However, many Arabic dialects, particularly those like Algerian Arabic, lack the same level of resources.

This exclusion limits access to education, healthcare, and business, reinforcing existing social and economic disparities. By narrowing the focus to Arabic, we aim to address this gap, starting from the high-resource MSA to the low-resource Arabic dialects, particularly Algerian Arabic, which is the primary focus of this study.

By addressing the resource gap, this research aims to develop more inclusive NLP systems that empower speakers of low-resource languages and enable their full participation in the global digital economy. Beyond practical applications, advancing NLP for these languages also contributes to the preservation of linguistic diversity and cultural heritage.

A promising approach to mitigating these challenges is leveraging high-resource languages, such as MSA, to support low-resource varieties. High-resource languages and their dialectal counterparts often share linguistic features, including core vocabulary, syntactic structures, and grammatical rules. These similarities provide a foundation for knowledge transfer, allowing techniques such as cross-lingual transfer learning, multitask learning, and feature fusion to improve NLP performance for low-resource languages. Focusing on Arabic dialects, this research will utilize MSA as a bridge to enhance NLP capabilities for Algerian Arabic.

This research is driven by the need to create equitable NLP systems that benefit all speakers, regardless of language. By exploring innovative, data-driven methodologies, this work aims to bridge the technological gap and ensure that AI-driven language technologies support linguistic

diversity rather than marginalize it. Instead of relying only on expert-crafted rules or manual annotation, which are often scarce or inconsistent for dialects like Algerian Arabic, this thesis uses computational models to learn directly from real-world text. While this approach enables scalability and adaptability, it also comes with limitations, such as potential data bias and reduced ability to capture cultural or pragmatic subtleties where the model might miss deeper meaning, context, or culturally specific language usage. These challenges are acknowledged and addressed through techniques that integrate linguistic proximity, cross-lingual knowledge transfer, and dialect-aware modeling.

1.3 Research Problem and Objectives

A core challenge in NLP is the resource disparity between MSA and Arabic dialects. While MSA benefits from extensive linguistic resources, Arabic dialects such as Algerian, Egyptian, Moroccan, and Tunisian Arabic remain significantly under-resourced. These dialects lack standardized orthographies, annotated datasets, and tailored NLP tools, making it difficult to develop accurate and scalable models for dialect identification, sentiment analysis, and other NLP tasks.

The linguistic divergence between MSA and dialects further complicates the problem. While MSA is a formal, standardized variety used in media, literature, and official communication, dialects are primarily spoken and exhibit substantial variation in vocabulary, grammar, and pronunciation across regions. This divergence makes it challenging to transfer knowledge from MSA to dialects, especially when dealing with informal, unstructured, and non-standardized text. Additionally, the limited availability of annotated dialectal data hinders the application of traditional supervised learning approaches, which rely heavily on large-scale labeled datasets.

The research problem, therefore, centers on how to effectively leverage the rich resources available for MSA to improve NLP tasks for low-resource Arabic dialects. This involves addressing challenges such as linguistic variation, and knowledge transfer between MSA and dialects.

This research aims to enhance NLP for low-resource Arabic dialects by leveraging knowledge from MSA, the specific objectives are:

1. To design and evaluate effective transfer learning approaches that adapt linguistic knowledge from MSA to low-resource Arabic dialects, using techniques such as cross-lingual transfer and feature fusion, with a focus on performance improvements over dialect-only baselines.

2. To improve dialect identification and sentiment analysis tasks by assessing the contribution of MSA-pretrained models and datasets to classification accuracy and sentiment prediction across a range of dialectal benchmarks.

3. To quantify the benefits of MSA-to-dialect knowledge transfer by systematically comparing models trained on dialectal data alone with those augmented by MSA knowledge, evaluating their generalization capabilities and robustness on unseen dialects.

4. To develop robust model architectures tailored to mitigate the effects of data scarcity and distributional imbalance, ensuring stable performance in low-resource and noisy dialectal environments.

5. To foster linguistic inclusivity in AI systems by investigating whether the integration of MSA knowledge reduces the disparity in performance between high-resource and low-resource dialects, thus contributing to more equitable language technology access across Arabic-speaking communities.

To address this gap, this research explores the hypothesis that knowledge acquired from MSA characterized by abundant annotated corpora and well-trained models can be transferred to improve the understanding of Algerian Arabic dialectal texts.

1.4 Research Questions

1. To what extent can models without any prior linguistic knowledge capture meaningful patterns in Arabic dialectal text, and does their performance justify the use of MSA a closely related high-resource variety as a source of transferable knowledge?

2. Can models trained exclusively on MSA generalize to dialectal Arabic despite lacking direct exposure to its linguistic patterns?

3. Can applying MSA-trained models to scenarios that require distinguishing between closely related language varieties provide evidence of their capacity to capture dialect-specific features despite having no direct exposure to them?

4. Can combining complementary representations such as deep contextual embeddings and subword-level information, improve the model's capacity to capture dialectal variation?

5. Can we use MSA data to support learning in dialectal settings where annotated data is scarce, by transferring linguistic knowledge across varieties?

6. Can learning from MSA and dialectal data simultaneously improve model's ability to capture generalizable linguistic features without overfitting to either variety?

7. Can shared representations learned from MSA data generalize beyond a single task, enabling adaptability to related tasks or other dialectal contexts?

1.5 Research Contributions

This PhD thesis contributes to advancing NLP for low-resource languages in the following ways:

1. **Hybrid Model for Arabic Dialect Identification** This work introduces a novel architecture that integrates CAMeLBERT and FastText embeddings for dialect classification. By combining contextual embeddings from CAMeLBERT with subword-level embeddings from FastText, the model enhances the identification of Arabic dialects, particularly Algerian Arabic, and provides insights into how embedding fusion can improve performance for low-resource dialects.

2. **Cross-Lingual Transfer Learning from MSA** The thesis demonstrates how transfer learning from MSA significantly improves performance on dialect identification tasks for low-resource dialects, reducing reliance on dialect-specific annotated data. By leveraging MSA-trained embeddings, this work shows that knowledge transfer from a high-resource language can enhance model performance on a low-resource dialect like Algerian Arabic.

3. **Multitask Learning for Low-Resource Sentiment Analysis** The thesis introduces SILAA-SA, a novel multitask learning framework designed to improve sentiment analysis in Algerian Arabic. This framework leverages MSA's Masked Language Modeling (MLM) to jointly learn from MSA data and Algerian dialect data, improving sentiment analysis accuracy on the low-resource Algerian dataset.

4. **MoE for Cross-Dialect Knowledge Sharing** A MoE layer is introduced, which dynamically selects relevant linguistic knowledge across MSA and dialectal data, improving model generalization. This layer allows the model to draw on both MSA and dialect-specific data during training, providing a more robust understanding of diverse linguistic features.

5. **Cross-Dialect Evaluation** Extensive experiments were conducted on different Arabic dialacts including Algerian, Tunisian, and Moroccan datasets to evaluate the generalizability of the proposed methods across Arabic dialects. The results demonstrate that leveraging MSA-based transfer learning significantly enhances performance across different dialects, highlighting the potential for cross-dialect knowledge sharing.

6. **State-of-the-Art Performance** The models developed in this thesis achieve superior results compared to traditional transfer learning approaches and existing transformer-based models. These results, demonstrated across multiple datasets such as MADAR, setting a new benchmark for Arabic dialect NLP tasks and contribute valuable insights into the application of MSA knowledge for dialect identification and sentiment analysis.

1.6 Thesis Structure

This thesis is organized into several chapters, each building upon the previous to develop a comprehensive understanding of the research problem and proposed solutions. The structure is designed to guide the reader from foundational concepts to advanced methodologies, experimental results, and conclusions. Below is an overview of each chapter:

Chapter II — Theoretical Background

This chapter explores the linguistic structure of Arabic, including its morphology, syntax, and dialectal variations, with a focus on the Algerian dialect. It then provides an overview of foundational NLP techniques, covering traditional machine learning approaches and advanced deep learning methods used for language processing. Finally, it discusses word representations, ranging from classical statistical methods to modern contextualized embeddings, highlighting their role in capturing semantic and syntactic relationships.

Chapter III — Literature Review

This chapter provides a comprehensive review of state-of-the-art approaches in Arabic Dialect Identification (ADI) and Sentiment Analysis, two tasks that exemplify the challenges of low-resource languages. The review categorizes methodologies into three paradigms: Traditional Machine Learning, Deep Learning (Non-Transformer), and Transformer-Based (Transfer Learning) approaches. It examines the progression from feature-based methods to advanced transformer models, highlighting key challenges such as morphological complexity, dialectal diversity, and fine-grained classification. The chapter critically analyzes representative works, emphasizing their strengths, limitations, and the evolution of the field. It concludes by identifying key gaps, and outlines how these insights guide the research trajectory of this thesis.

Chapter IV — Exploratory Analysis of Classical and Transfer Learning for Dialectal NLP

This chapter provides complementary insights into Arabic dialect NLP through two key experiments. First, a disaster classification study demonstrates that traditional Machine Learning (ML) models, despite lacking pretrained linguistic knowledge, can effectively capture the statistical patterns of Arabic dialectal text, establishing a strong baseline for time-critical tasks. Second, an investigation into MSA-based transfer learning reveals that leveraging linguistic knowledge from MSA yields competitive results, closely rivaling specialized models like DziriBERT. While MSA-based pretrained models do not significantly outperform dialect-specific models, their performance highlights the potential of exploiting the linguistic proximity between MSA and dialects like Algerian Arabic. These findings suggest that integrating MSA knowledge—a resource-rich

linguistic source—can enhance performance in complex NLP tasks, offering scalable and generalizable solutions. The chapter concludes by motivating further research into hybrid approaches that combine MSA-based pretrained representations with advanced model architectures, aiming to surpass specialized models and set new benchmarks in Arabic dialect NLP.

Chapter V — Cross-Lingual Dialect Identification using Hybrid Architectures: The WASL-DI Approach

This chapter introduces WASL-DI, a hybrid cross-lingual model for ADI with a focus on low-resource dialects like Algerian Arabic. The model combines CAMeLBERT for deep contextual representations and FastText for subword-level features, addressing challenges such as data scarcity, dialectal diversity, and the lack of standardized orthography. This dual-path architecture processes input through both embeddings, fuses the outputs, and classifies them via a Multi-Layer Perceptron (MLP). Experiments on the MADAR dataset demonstrate that WASL-DI achieves 99.24% accuracy, outperforming traditional ML methods and specialized transformer models like DziriBERT and MDA-BERT, even with limited training data. Robustness tests on noisy and incomplete data confirm the model's resilience, with only a minor drop in performance across diverse Arabic dialects. An ablation study highlights the critical contribution of combining CAMeLBERT and FastText, showing that each component is essential for superior performance. The analysis also reveals the model's sensitivity to script variations, particularly its association of Arabizi (Latin script for Arabic) with the Algerian dialect—a factor that serves as both a strength and a limitation. WASL-DI successfully leverages MSA resources to enhance dialect identification, providing a scalable and adaptable framework for low-resource NLP tasks. This chapter lays the groundwork for future research in cross-lingual transfer learning, promoting more inclusive and accessible language technologies.

Chapter VI — Multitask Learning for Sentiment Analysis in Algerian Arabic: A Transfer-Based Framework SILAA-SA

This chapter presents SILAA-SA, a multitask learning framework developed to improve sentiment analysis in low-resource dialects, with a primary focus on Algerian Arabic. Unlike traditional transfer learning, SILAA-SA jointly learns two tasks: sentiment analysis on dialectal data and MLM on MSA, utilizing a shared MoE layer enhanced with self-attention. This design enables the model to capture and transfer shared linguistic features between MSA and dialects. Experimental results confirm SILAA-SA's effectiveness across multiple datasets: it achieves high performance on Algerian (FASSILA), Tunisian (TSAC), Moroccan (MAC, MYC), and other multi-dialect dataset (arSarcasmV2) corpora, and also demonstrates strong generalization to a separate taskfake news detection. Comprehensive ablation studies further validate the contribution of each architectural component. By strategically incorporating MSA resources as data, SILAA-SA offers a robust, scalable solution for addressing the limitations of low-resource Arabic NLP. This chapter builds

directly on the MSA transfer approach discussed in Chapter IV and the hybrid modeling strategies introduced in Chapter V, while establishing multitask learning as a new and promising paradigm for inclusive and adaptable language technologies.

1.7 Publications

This thesis investigates two central NLP tasksdialect identification and sentiment analysisthrough three experimental phases (Chapters 4, 5, 6). The research is grounded in a series of peer-reviewed publications authored by the thesis author, including a journal paper, a manuscript currently under review, and two conference proceedings. These publications demonstrate both the real-world applicability and the novelty of the proposed approaches.

1. M. Chabane, F. Harrag, and K. Shaalan, "Advancing low-resource dialect identification: A hybrid cross-lingual model leveraging CAMeLBERT and FastText for Algerian Arabic," *Expert Systems with Applications*, vol. 284, 2025, Art. no. 127816, doi: 10.1016/j.eswa.2025.127816.

2. M. Chabane, F. Harrag, K. Shaalan, and S. Hamdi, "Bridging the Gap: Transfer Learning for Dialect Identification in Low-Resource Settings A Case Study with Algerian Arabic," *2025 International Symposium on iNnovative Informatics of Biskra (ISNIB)*, Biskra, Algeria, 2025, pp. 1–6, doi: 10.1109/ISNIB64820.2025.10982839.

3. M. Chabane, F. Harrag, and K. Shaalan, "Beyond Deep Learning: A Two-Stage Approach to Classifying Disaster Events and Needs," *2024 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pp. 1–7, 2024, doi: 10.1109/ICT-DM62768.2024.10798928.

4. M. Chabane, F. Harrag, and K. Shaalan, (2025). SILAA-SA: A Multitask Mixture-of-Experts Framework for Sentiment Analysis in Low-Resource Arabic Dialects via Modern Standard Arabic Transfer. IEEE Transactions on Affective Computing. Under review.

CHAPTER 2

THEORETICAL BACKGROUND

This chapter lays the foundational groundwork necessary for understanding the research presented in this thesis. It introduces the linguistic structure and diversity of the Arabic language, with a particular emphasis on the Algerian dialect. It also provides a comprehensive overview of traditional and modern NLP techniques, including machine learning, deep learning, and word representations, which are pivotal to the design and evaluation of the proposed approaches.

Arabic, a linguistically rich and diverse language, exists in three primary forms, each serving distinct purposes rooted in history, contemporary communication, and regional variation, respectively. This chapter begins by exploring the linguistic structure of Arabic, including its morphology, syntax, and phonology, as well as the complexities of dialectal variation, with a particular focus on the Algerian dialect a low-resource language that has only recently gained attention in the NLP community. Beyond Arabic linguistics, the chapter delves into foundational NLP techniques, starting with traditional machine learning approaches, which provide a strong basis for understanding and solving NLP tasks. It then transitions to advanced deep learning models, and the revolutionary Transformer architecture, which have significantly enhanced the ability of NLP systems to handle sequential data, capture long-range dependencies, and generate context-aware representations. Finally, the chapter discusses word representations, from classical methods to modern word embeddings, as well as contextualized embeddings, which have become the cornerstone of modern NLP by enabling dynamic, context-sensitive representations of words.

2.1 Arabic Linguistics and Dialectology

Arabic stands out among the world's languages for its intricate structure, rich historical legacy, and remarkable diversity [44]. It is represented in three distinct forms: Classical Arabic (CA), the

ancestral language spoken over fourteen centuries ago; MSA, a dynamic and evolving language that incorporates new words and expressions to meet the needs of contemporary speakers; and Dialectal Arabic (DA), which encompasses a diverse range of regional dialects. As one of the principal Semitic languages, Arabic not only underpins a vast literary and religious heritage but also exhibits unique linguistic phenomena that continue to evolve in modern communication. In this section, we introduce the essential elements of Arabic's linguistic structure and its varied dialects. We then discuss the challenges that arise when applying NLP techniques to Arabic, with a particular focus on the complexity of dialect processing.

2.1.1 Linguistic Structure of Arabic and Dialectal Variation

Arabic is a linguistically rich and diverse language, with a complex structure that includes notable differences between MSA and its regional dialects. These differences affect vocabulary, morphology, phonetics, and syntax, which not only shape how Arabic is spoken but also introduce significant challenges for NLP systems.

2.1.1.1 Morphology

At the core of Arabic morphology lies the root-and-pattern system, a non-linear mechanism that distinguishes it from many Indo-European languages. Words are typically derived from triliteral (three-consonant) or quadriliteral (four-consonant) roots, combined with vocalic and affixal patterns to generate semantically related forms. For example, the root كَتَبَ -kataba- (Wrote) produces words such as كِتَابٌ -kitab- (book), مَكْتَبَةٌ -maktaba- (Library), and مَكْتَبْ -maktab- (Office). This system enriches the lexicon but also complicates computational modeling due to its non-concatenative nature.

Additionally, Arabic exhibits extensive inflectional complexity, where a single root can yield numerous derived forms influenced by gender, number, tense, and mood [73]. This variability poses challenges for morphological segmentation, lemmatization, and part-of-speech tagging, requiring computational models capable of handling both derivational and inflectional variations.

2.1.1.2 Syntax

Arabic syntax is highly flexible, with CA favoring a Verb-Subject-Object (VSO) order, while modern usage especially in dialectscommonly follows a Subject-Verb-Object (SVO) structure. This syntactic variation influences semantic emphasis and sentence structure, making it essential for NLP systems to dynamically adapt to it.

Moreover, Arabic enforces strict grammatical agreement in gender, number, and case among sentence elements. Adjectives must align with the nouns they modify, and verbs must reflect the subject's plurality and gender [38]. Parsing algorithms and grammatical analyzers must therefore

incorporate sophisticated rule-based or machine learning approaches to correctly capture these dependencies.

2.1.1.3 Phonology

Arabic has unique sounds, including emphatic consonants that are pronounced with a heavier quality in the throat. The language also distinguishes between short and long vowels, which can change the meaning of words and impact speech processing [6].

This phonetic structure exhibits substantial variation between MSA and dialects, especially in terms of the articulation of consonants, vowels, and prosodic features. For example different Arabic dialects pronounce consonant in varying ways.

- The pronunciation of the letter ق-qaf- varies significantly across Arabic dialects. In many Levantine dialects, it is realized as a glottal stop, while in Gulf and Egyptian dialects, it is often pronounced as a hard "g" (as in "go"). Within the Algerian dialect alone, this variation is evident: the central region predominantly uses the original ق-qaf- sound, whereas the eastern and western regions favor the hard "g" pronunciation.

- The letter ج -jim- is pronounced like "j" in judge in most dialects but as "g" in go in Egyptian Arabic.

Dialectal differences extend to stress patterns, pitch contours, and intonation as they vary across dialects. These elements are crucial in spoken language as they affect the rhythm, melody, and meaning of an utterance, this can influence how questions, commands, or statements are interpreted, requiring NLP systems to adapt to these contextual features. These prosodic differences pose a significant challenge in developing models that can handle all dialects equally well.

2.1.1.4 Arabic Dialects from a Sociolinguistic Perspective

Beyond the structural differences discussed above, Arabic variation can also be analyzed through well-established sociolinguistic theories that provide crucial insight into dialect usage and its implications for NLP. One of the foundational concepts in Arabic sociolinguistics is Ferguson's theory of diglossia [79], which explains the coexistence of a "high" (H) variety (MSA) and "low" (L) varieties (regional dialects like Algerian Arabic). According to Ferguson, MSA is typically reserved for formal domains such as education, religion, and media, while dialects are used in everyday, informal communication. This distinction introduces complexities in NLP, as the models must navigate lexical, morphological, and syntactic disparities between the H and L varieties.

Complementing this, Labov's theory of variation [111] highlights how linguistic differences are influenced by social factors such as age, gender, region, and socioeconomic status. In the

context of Algerian Arabic, such variation is reflected in accent, vocabulary, and even orthographic conventions (e.g., Arabizi). From a computational perspective, this implies that dialectal NLP systems must be robust to intra-dialect variation and capable of generalizing across speaker populations.

Furthermore, code-switching and lexical borrowing (discussed in detail in the following section) are directly related to language contact theory. In multilingual settings like Algeria, linguistic boundaries are porous, with frequent mixing of French, Arabic, and Berber. This results in hybrid utterances that challenge conventional NLP pipelines. Handling these phenomena effectively requires models that support mixed-language inputs and can leverage cross-lingual embeddings.

Among the many regional varieties, Algerian Arabic presents a particularly complex and under-resourced case, warranting special attention in NLP research.

2.1.2 Algerian dialect

Algerian Arabic, also known as Darija, is a prominent dialect spoken by millions of people in Algeria, as well as by Algerian communities abroad. Despite its widespread use, the dialect has historically been considered a low-resource language, and only in recent years has it gained some minor attention in the field of NLP. For much of its history, Algerian Arabic was excluded from mainstream linguistic studies and computational resources, often being overshadowed by MSA, which has a more formal and standardized structure. This absence of resources was especially notable until just over a decade ago when the language was still classified as a non-resourced language [119].

The situation of Algerian Arabic is not unique to this dialect alone, as many other regional Arabic dialects face similar challenges. However, its unique characteristics shaped by a rich history of linguistic interactions with Berber, French, and other regional languages have added to the complexity of creating standardized computational models. Algerian Arabic is a highly dynamic and fluid dialect, with significant regional variation in terms of vocabulary, phonology, and syntax. This variability poses a major obstacle for NLP systems that aim to process the language effectively.

Several other factors other than lack of resources contribute to the complexity of developing robust NLP models for Algerian Arabic:

- **Lack of standardization :** Algerian Arabic lacks a standardized written form. It's primarily spoken, with no formalized orthography. When written, it may either appear in normal script or in Arabizi, a blend of Latin script and numbers representing Arabic phonetics. As a result, the same words can be written in multiple ways, leading to inconsistencies in data processing. For example, the sentence "رايح لسطيف" which means "I am going to Setif" Can be written in different froms:

  - "Raye**h** l Setif," where the "h" corresponds to "ح".
  - "Raye**7** l Setif," where the "7" in Algerian Arabizi represents "ح".
  - "Rayeh l **Stif**," where "Stif" is written phonetically instead of using the formal name of the city.
  - "Rayeh l S**6**if," where the "6" signifies "ط".

- **Code-Switching :** Algerian Arabic frequently incorporates elements from other languages, particularly French, in a phenomenon known as code-switching. This happens seamlessly within sentences, reflecting the country's multilingual heritage.

  For instance, the sentence "الكواغط فوق الطابلة" which means "the papers are above the table" has three words, each from a different language:

  - "الكواغط" (Papers) Derived from the Turkish word "kâğıt", this word has been adapted to Algerian Arabic with the plural suffix applied according to Algerian dialect rules, not Turkish ones.
  - "فوق" (Above) This is a purely Arabic word.
  - "الطابلة" (The table) originating from the French word "la Table", this word has been feminized according to Arabic rules by adding the feminine suffix "ة" and the definite article "ال", equivalent to "la" in French or "the" in English.

- **Dialectal Variation :** Algerian Arabic exhibits significant dialectal variation across different regions, with distinct differences in vocabulary, pronunciation, and especially accent.

- **Intonation-Based Question Identification :** A distinctive feature of Algerian Arabic is its use of intonation to differentiate between statements and questions. Unlike many other languages, Algerian Arabic does not always rely on explicit question words (like "is" or "are" in English) or auxiliary verbs to form questions. Instead, the intent of a sentence is conveyed primarily through the speaker's intonation.

  For example, the sentences "He is going home/ is he going home?" can be expressed in Algerian Arabic as "او رايح للدار". Whether this sentence is a statement or a question depends solely on the intonation used. If the sentence is spoken with a special intonation at the end, it becomes a question. If not, it is a statement.

- **Borrowed Words and Their Adaptation :** Algerian Arabic has borrowed numerous words from other languages and adapted them according to its own phonological and morphological rules.

  For instance, the French word "table" in plural becomes "twabal" in Algerian Arabic, and the Turkish word "kâğıt" is "kaghet" with its plural form "kwaghet" instead of the Turkish "kâğıtlar.".

Table 2.1 presents examples of words borrowed into Algerian Arabic from Turkish and French, showing how they are adapted in pronunciation and form to fit the dialect's phonological and morphological rules.

English	**Algerian dialect**	**Origin word**	**Origin language**
Maybe	بالاك	Belki	Turkish
Poor	زوالي	Zavallı	
Paper	كاغط	kâğıt	
Cork	بوشون	Bouchon	French
Casserole	كوكوطة	Cocotte	
Wheelbarrow	برويطة	Brouette	

Table 2.1: Sample of words borrowed into Algerian dialect

2.2 Traditional Machine Learning Approaches

NLP has been significantly influenced by traditional machine learning approaches. These methods, while often simpler than modern deep learning techniques, provide a strong foundation for understanding and solving NLP tasks. This section explores some of the most widely used traditional machine learning algorithms in NLP, including Decision Trees, Random Forests (RF), Naïve Bayes (NB), Extreme Gradient Boosting (XGBoost), and Support Vector Machine (SVM).

2.2.1 Decision Trees

Decision Trees [58] are a widely used non-parametric supervised learning method for both classification and regression tasks. They operate by recursively partitioning the dataset based on input features, ultimately forming a tree-like structure where each internal node represents a decision, branches correspond to possible outcomes, and leaf nodes denote final predictions.

The process of constructing a Decision Tree involves selecting the most informative features at each step to maximize predictive performance. The selection of the best feature for splitting is crucial and relies on specific criteria such as **Information Gain** and **Gini Impurity** [85].

Entropy quantifies the impurity or uncertainty in a dataset and is defined as:

$$H(D) = -\sum_{i=1}^{C} p_i \log_2 p_i \tag{2.1}$$

where p_i is the probability of class i in dataset D, and C is the total number of classes. A higher entropy value indicates greater disorder in the dataset.

Information Gain is based on entropy reduction and is calculated as follows:

$$IG(D,A) = H(D) - \sum_{v \in \text{Values}(A)} \frac{|D_v|}{|D|} \times H(D_v) \tag{2.2}$$

where $H(D)$ represents the entropy of dataset D, and D_v denotes the subset where feature A has a specific value v. A higher Information Gain indicates a more informative feature for splitting.

Gini Impurity, another widely used splitting criterion, measures the impurity of a node and is given by:

$$Gini(D) = 1 - \sum_{i=1}^{C} p_i^2 \tag{2.3}$$

where p_i is the probability of class i in dataset D, and C is the total number of classes. A lower Gini Impurity indicates a purer node.

Despite their interpretability and ease of use, Decision Trees have notable limitations. They are highly susceptible to overfitting, particularly when dealing with complex datasets or deep trees with numerous splits. Regularization techniques such as pruning, setting a maximum depth, or using ensemble methods like Random Forests can help mitigate these issues.

2.2.2 Naïve Bayes

Naïve Bayes is a probabilistic classifier based on Bayes' theorem, assuming that features are conditionally independent given the class label. Despite its simplicity, Naïve Bayes performs well in many real-world applications.

Bayes' theorem states:

$$P(C_k \mid X) = \frac{P(X \mid C_k)P(C_k)}{P(X)} \tag{2.4}$$

where:

- $P(C_k \mid X)$ is the posterior probability of class C_k given feature vector X.
- $P(X \mid C_k)$ is the likelihood of X given class C_k.
- $P(C_k)$ is the prior probability of class C_k.
- $P(X)$ is the prior probability of X.

Under the naive assumption that all features x_i are independent given C_k, the likelihood can be factorized as:

$$P(X \mid C_k) = \prod_{i=1}^{n} P(x_i \mid C_k) \tag{2.5}$$

which simplifies computation significantly.

Naïve Bayes is computationally efficient and performs well when the independence assumption holds, but its performance degrades when features are strongly correlated.

2.2.3 Support Vector Machine

SVM [64] is a powerful supervised learning model used for classification and regression tasks. The main objective of an SVM is to find an optimal hyperplane that best separates data points of different classes in a high-dimensional space. The decision boundary in an SVM is defined by the equation:

$$w \cdot x + b = 0 \tag{2.6}$$

where w is the weight vector, x represents the input features, and b is the bias term.

To ensure that the model generalizes well, SVMs aim to maximize the margin between the closest data points (support vectors) of each class. The margin is defined as:

$$\text{Margin} = \frac{2}{\|w\|} \tag{2.7}$$

This margin should be as large as possible, subject to the constraint that all data points satisfy the inequality:

$$y_i(w \cdot x_i + b) \geq 1, \quad \forall i \tag{2.8}$$

where $y_i \in \{-1, 1\}$ is the class label.

SVMs are widely used in text classification, image recognition, and bioinformatics due to their ability to handle high-dimensional data and robustness against overfitting.

2.2.4 Ensemble Learning

Ensemble learning refers to the technique of combining multiple models, or base learners, to produce a stronger overall model. The primary goal is to improve the model's accuracy, robustness, and generalization by leveraging the strengths of various learning algorithms.

2.2.4.1 Bagging

Bagging [56], short for Bootstrap Aggregating, is an ensemble method designed to improve the accuracy and stability of machine learning algorithms. It works by generating multiple subsets of the data through bootstrap sampling, where each subset is sampled randomly with replacement from the original dataset. A model is trained on each subset, and the final prediction is made by

aggregating the predictions of all individual models. Bagging is particularly effective at reducing variance and mitigating overfitting.

Let $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be the original dataset, and D_b be the bootstrap sample for the b^{th} model. The aggregate prediction of the ensemble model can be expressed as:

$$\hat{y}_{\text{bagging}} = \frac{1}{B} \sum_{b=1}^{B} \hat{y}_b \tag{2.9}$$

where $\hat{y}_b$ is the prediction from the b^{th} model, and B is the total number of base models. By aggregating the predictions from multiple models, bagging reduces variance and increases predictive performance.

2.2.4.2 Random Forest

Random Forest [57] is an extension of the Bagging method, designed to enhance model accuracy by reducing variance and preventing overfitting. It achieves this by constructing an ensemble of decision trees, each trained on a different bootstrap sample of the dataset. Additionally, at each decision tree node, only a random subset of features is considered, which decreases the correlation between individual trees and ensures greater model diversity. This inherent randomness strengthens the robustness of the Random Forest model.

The final prediction in Random Forest is obtained by aggregating the predictions from all trees. For classification tasks, majority voting is employed, while for regression, predictions are averaged. Mathematically, the predictions are defined as:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^{B} T_b(x) \tag{2.10}$$

where $T_b(x)$ represents the prediction of the b^{th} tree, and B denotes the total number of trees in the forest.

Random Forest's key advantages include its robustness to noisy data, the ability to handle missing values, and its efficiency with high-dimensional datasets. Moreover, it generally performs well with minimal hyperparameter tuning, making it a popular choice for a wide range of machine learning tasks.

2.2.4.3 Boosting

Boosting [81] is an ensemble learning technique where multiple weak models are trained sequentially, with each iteration adjusting the weights of training samples to emphasize those that were previously misclassified. Let D represent the training dataset of size n, and let w_i be the weight assigned to the i-th example.

During training, m models are iteratively trained, denoted as $f_i(x)$, using the weighted training data based on w_i. After each iteration, these weights are updated through a weighting function $W(y, f_i(x))$, which depends on the actual label y and the predicted value $f_i(x)$. The final prediction is determined by computing a weighted sum of the individual models:

$$f_{\text{boosting}}(x) = \sum_{i=1}^{m} \alpha_i f_i(x) \tag{2.11}$$

where α_i represents the weight assigned to the i-th model. This weight is computed as the logarithm of the ratio between the model's error and the error of a random classifier:

$$\alpha_i = \log \frac{1 - \text{error}_i}{\text{error}_i} \tag{2.12}$$

where error_i denotes the error of the i-th classifier, which is derived from the sum of the weights of the misclassified examples.

2.2.4.4 Extreme Gradient Boosting

XGBoost [61] is a highly efficient and scalable implementation of the boosting algorithm. XGBoost incorporates regularization, handles missing values, and uses parallel processing for faster training. It constructs models sequentially by adding new learners to minimize the residual errors from previous models. The key innovation in XGBoost is the use of second-order gradient information, which enhances optimization and leads to faster convergence.

The optimization problem in XGBoost is defined by the following objective function:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t) \tag{2.13}$$

where:

- $l(y_i, \hat{y}_i)$ is the loss function that measures the difference between the true label y_i and the predicted value $\hat{y}_i$,
- $\Omega(f_t)$ is the regularization term for the complexity of the trees, which helps prevent overfitting,

The regularization term is given by:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_j w_j^2 \tag{2.14}$$

where:

- T is the total number of trees in the model.
- γ controls the complexity of the model by penalizing the number of trees,
- λ is the L2 regularization term on leaf weights w_j.

2.2.4.5 Voting Classifier

A Voting Classifier is a simple ensemble method [68] that combines multiple classification models and predicts the class label based on majority voting. Each base classifier in the ensemble predicts a class label, and the final class label is determined by the most common prediction. Voting Classifiers can be used with any type of classifier and typically perform well by leveraging the strengths of different base learners.

The prediction from a Voting Classifier can be made using either **hard voting** or **soft voting**:

- **Hard Voting:** The final prediction is determined by majority rule, where each individual model in the ensemble casts a vote for a class, and the class with the most votes is selected. Mathematically, it can be expressed as:

$$\hat{y}_{\text{hard}} = \arg\max_k \sum_{b=1}^{B} \mathbb{1}(T_b(x) = k) \tag{2.15}$$

  where $T_b(x)$ represents the prediction of the b^{th} model, and B is the total number of models in the ensemble.

- **Soft Voting:** Instead of selecting the class with the most votes, soft voting averages the predicted probabilities from all models and selects the class with the highest probability:

$$\hat{y}_{\text{soft}} = \arg\max_k \sum_{b=1}^{B} w_b P_b(y = k \mid x) \tag{2.16}$$

  where $P_b(y = k \mid x)$ is the probability predicted by the b^{th} model for class k, and w_b is an optional weight assigned to model b.

This method is effective when combining models with complementary strengths, as it tends to produce more stable and accurate predictions than any individual model in the ensemble.

2.3 Deep Learning

Deep learning is a subfield of machine learning that focuses on training artificial neural networks with multiple layers to model complex patterns in data. These neural networks are inspired by the structure and function of the human brain, consisting of interconnected nodes (or neurons) that process and transmit information. Deep learning has revolutionized many domains, including computer vision, speech recognition, and NLP, due to its ability to automatically learn hierarchical representations from raw data.

2.3.1 Deep Neural Networks

A neural network is composed of layers of neurons, each of which performs a simple computation. The three primary types of layers are:

- **Input Layer**: This layer receives the raw input data (e.g., word embeddings, pixel values, etc.).
- **Hidden Layers**: These layers transform the input data through a series of nonlinear operations. Deep networks contain multiple hidden layers, enabling them to learn increasingly abstract features.
- **Output Layer**: This layer produces the final prediction or classification, such as a probability distribution over possible classes.

Each neuron in a layer is connected to neurons in the subsequent layer via weighted connections. The output of a neuron is computed as a weighted sum of its inputs, followed by the application of an activation function, such as the Rectified Linear Unit (ReLU) or sigmoid function.

2.3.1.1 Forward Propagation

Forward propagation is the process by which input data is passed through the network to generate an output. For a given input $\mathbf{x}$, the output of each layer is computed as follows:

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} \tag{2.17}$$

$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)}) \tag{2.18}$$

where:

- $\mathbf{z}^{(l)}$ is the weighted input to layer l,

- $\mathbf{W}^{(l)}$ is the weight matrix for layer l,
- $\mathbf{a}^{(l-1)}$ is the activation from the previous layer,
- $\mathbf{b}^{(l)}$ is the bias vector for layer l,
- f is the activation function.

The final output $\mathbf{a}^{(L)}$ of the network is compared to the true label using a loss function $\mathscr{L}$, which quantifies the error between the prediction and the ground truth.

2.3.1.2 Backpropagation

Backpropagation is the core algorithm used to train neural networks. It involves computing the gradient of the loss function with respect to each weight in the network, allowing the weights to be updated in a way that minimizes the loss. The process consists of the following steps:

1. **Forward Pass** : As described in Section 2.3.1.1, compute the network's output and the loss by passing the input data through the network.
2. **Backward Pass**: Compute the gradient of the loss with respect to each weight using the chain rule of calculus.
3. **Weight Update**: Adjust the weights using gradient descent or a variant (e.g., Adam, RMSProp) to minimize the loss.

The gradient for a weight $w_{ij}^{(l)}$ in layer l is computed as:

$$\frac{\partial \mathscr{L}}{\partial w_{ij}^{(l)}} = \frac{\partial \mathscr{L}}{\partial z_i^{(l)}} \cdot \frac{\partial z_i^{(l)}}{\partial w_{ij}^{(l)}} \tag{2.19}$$

where $\frac{\partial \mathscr{L}}{\partial z_i^{(l)}}$ is the error term propagated backward from the subsequent layer.

2.3.1.3 Activation Functions

Activation functions introduce nonlinearity into the network, enabling it to learn complex patterns. Common activation functions include:

- **ReLU**: $f(z) = \max(0, z)$
- **Sigmoid**: $f(z) = \frac{1}{1+e^{-z}}$
- **Hyperbolic Tangent (tanh)**: $f(z) = \tanh(z)$
- **Softmax**: $\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}$

2.3.1.4 Training Neural Networks

Training a neural network involves iteratively updating its weights to minimize the loss function. Key considerations include:

- **Loss Function**: Measures the discrepancy between predictions and true labels (e.g., cross-entropy loss for classification, mean squared error for regression).
- **Optimization Algorithms**: Techniques like Stochastic gradient descent (SGD), Adam, and RMSProp are used to update weights efficiently.
- **Regularization**: Methods like dropout and weight decay prevent overfitting by encouraging the network to learn robust features.

Below is a pseudo-code summarizing the training process of a neural network:

Algorithm 1 Neural Network Training

Initialize weights $\mathbf{W}$ and biases $\mathbf{b}$ randomly
for epoch = 1 to N **do**
  for each batch of data $(\mathbf{X}, \mathbf{y})$ **do**
    Forward Pass:
    Compute activations $\mathbf{a}^{(l)}$ for each layer l using $\mathbf{a}^{(l)} = f(\mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)})$
    Compute loss $\mathscr{L}$ using the output $\mathbf{a}^{(L)}$ and true labels $\mathbf{y}$
    Backward Pass:
    Compute gradients $\frac{\partial \mathscr{L}}{\partial \mathbf{W}^{(l)}}$ and $\frac{\partial \mathscr{L}}{\partial \mathbf{b}^{(l)}}$ for each layer l using backpropagation
    Update weights and biases using an optimization algorithm (e.g., SGD, Adam):
    $\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \eta \frac{\partial \mathscr{L}}{\partial \mathbf{W}^{(l)}}$
    $\mathbf{b}^{(l)} \leftarrow \mathbf{b}^{(l)} - \eta \frac{\partial \mathscr{L}}{\partial \mathbf{b}^{(l)}}$
  end for
end for

2.3.2 Recurrent Neural Networks

Recurrent Neural Network (RNN) are a class of neural networks designed for sequential data processing. Unlike feedforward networks, RNNs incorporate hidden states that enable them to retain information from previous time steps.

Given an input sequence $X = \{x_1, x_2, \ldots, x_T\}$, an RNN updates its hidden state h_t at each time step t based on the previous hidden state h_{t-1} and the current input x_t:

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b_h) \tag{2.20}$$

where W_h and W_x are weight matrices, b_h is a bias term, and σ is an activation function, typically tanh or ReLU. The output y_t at each time step is computed as:

$$y_t = \phi(W_y h_t + b_y) \tag{2.21}$$

where W_y and b_y are the output weights and bias, and ϕ is the activation function.

However, standard RNNs suffer from the vanishing and exploding gradient problems [87], which hinder their ability to capture long-term dependencies. To address these issues, advanced architectures such as Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been developed.

2.3.2.1 Long Short-Term Memory

LSTM [89] networks extend traditional RNNs to effectively model long-range dependencies. Standard RNNs suffer from the vanishing gradient problem, making it difficult to capture dependencies over long sequences. LSTMs address this issue through a dedicated memory cell that maintains information across time steps, regulated by gating mechanisms.

The behavior of an LSTM unit at each time step t is governed by the following:

Forget Gate: The forget gate decides which part of the previous memory cell state c_{t-1} should be retained. It takes as input the current input vector x_t and the previous hidden state h_{t-1}, applies a sigmoid activation function, and produces a value between 0 and 1 for each memory cell unit:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2.22}$$

where W_f and U_f are learned weight matrices, b_f is a bias term, and σ is the sigmoid activation function.

Input Gate: The input gate determines how much new information should be stored in the memory cell. It has two components: the input gate activation i_t and the candidate memory update $\tilde{c}_t$. The activation is computed as:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2.23}$$

The candidate memory update $\tilde{c}_t$ represents new information that could be added to the cell state:

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{2.24}$$

Cell State Update: The new cell state c_t is a combination of the previous cell state c_{t-1}, scaled by the forget gate, and the newly computed candidate memory update, scaled by the input gate:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{2.25}$$

where $\odot$ represents element-wise multiplication.

Output Gate: Finally, the output gate determines how much of the updated cell state contributes to the next hidden state h_t, which is passed to the next time step:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{2.26}$$

Hidden State Update: The final hidden state h_t is obtained by applying the output gate activation to the cell state:

$$h_t = o_t \odot \tanh(c_t) \tag{2.27}$$

The hidden state h_t serves as the output of the LSTM unit and is used in subsequent computations.

Figure 2.1 illustrates the architecture of the LSTM unit, highlighting the different gates.

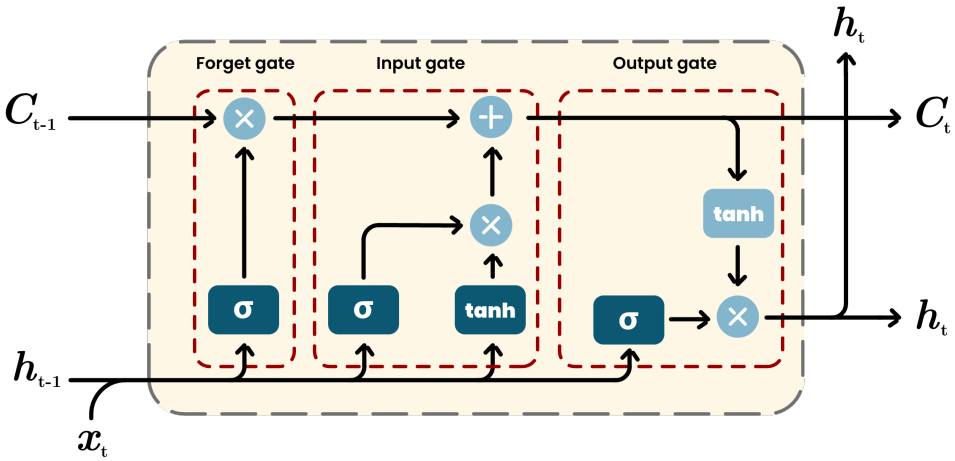

Figure 2.1: Illustration of the LSTM architecture, showing the gating mechanisms.

2.3.2.2 Bidirectional LSTM

A limitation of standard LSTMs is that they process sequences in a unidirectional manner, capturing only past dependencies. However, many sequence processing tasks require both past and future context. Bidirectional Long Short-Term Memory (BiLSTM)s address this by incorporating

two separate LSTMs: one processing the input from left to right (forward direction) and another from right to left (backward direction). The outputs of both LSTMs are then combined, typically via concatenation:

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{2.28}$$

where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ denote the hidden states from the forward and backward LSTM layers, respectively, and $\oplus$ represents concatenation.

2.3.2.3 Gated Recurrent Unit

GRUs [63] are a variant of RNNs designed to address the vanishing gradient problem while being computationally more efficient than LSTM networks. Unlike LSTMs, which use three separate gates, GRUs simplify the architecture by combining the forget and input gates into a single update gate. This reduction in complexity results in fewer parameters while maintaining comparable performance.

The operations in a GRU unit are governed by the following equations:

Update Gate: The update gate z_t decides the proportion of the previous hidden state h_{t-1} that should be retained in the current hidden state h_t. A higher value of z_t allows more past information to be preserved, while a lower value encourages the model to incorporate new information from x_t:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{2.29}$$

where W_z and U_z are learned weight matrices, b_z is a bias term, and σ represents the sigmoid activation function.

Reset Gate: The reset gate r_t determines how much of the past hidden state h_{t-1} should be forgotten before computing the candidate activation $\tilde{h}_t$. This enables the model to selectively reset parts of the memory when processing new information:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{2.30}$$

Candidate Activation: The candidate activation $\tilde{h}_t$ represents the new hidden state candidate. It is computed based on the current input x_t and the previous hidden state h_{t-1}, but only after applying the reset gate r_t. This allows the GRU to reset certain dimensions of h_{t-1} when needed:

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{2.31}$$

where $\odot$ represents element-wise multiplication.

Hidden State Update: The final hidden state h_t is a linear interpolation between the previous hidden state h_{t-1} and the newly computed candidate activation $\tilde{h}_t$, controlled by the update gate z_t:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{2.32}$$

This equation ensures that if z_t is close to 1, the new hidden state is primarily based on the candidate activation $\tilde{h}_t$, incorporating new information. Conversely, if z_t is close to 0, the previous hidden state h_{t-1} is largely retained.

The architecture of the GRU, including the update and reset gates, is illustrated in Figure 2.2.

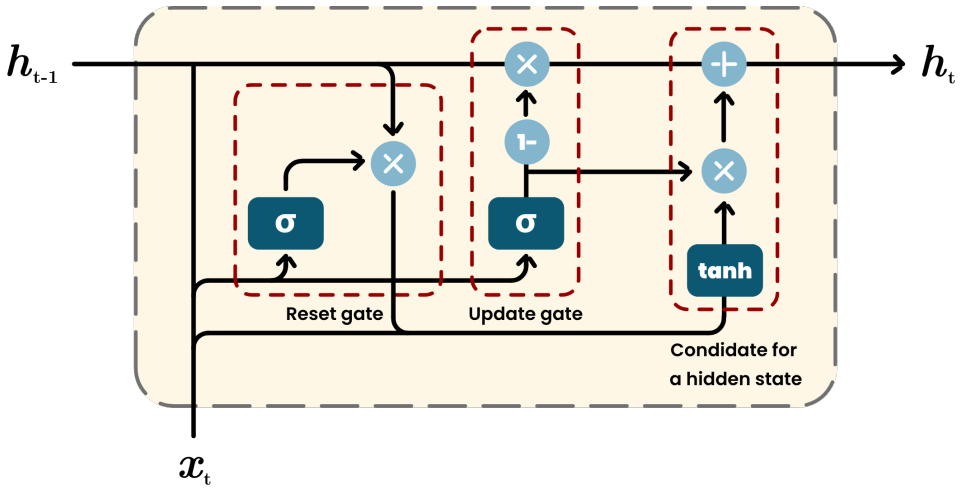

Figure 2.2: Illustration of the GRU architecture, showing the update and reset gates.

2.3.2.4 Bidirectional Gated Recurrent Unit

Similar to BiLSTM mentioned in section 2.3.2.2, a Bidirectional Gated Recurrent Unit (BiGRU) consists of two GRU layers that process the sequence in opposite directionsone forward and one backward. This allows the model to capture both past and future dependencies, leading to richer contextual representations. At each timestep, the hidden states from both directions are combined, typically through concatenation:

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{2.33}$$

where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are the forward and backward hidden states, respectively.

2.3.3 Convolutional Neural Networks

Convolutional Neural Network (CNN), a specialized type of neural network architecture designed to process data with grid-like structures, such as images, time-series data, or text. Inspired by the visual cortex in the brain, CNNs are particularly effective at capturing local patterns and spatial hierarchies in data. They were first introduced by Y. LeCun et al. [112] and have since become a cornerstone of modern deep learning.

Compared to traditional Deep Neural Networks (DNNs), CNNs exhibit two main characteristics:

- **Local Connectivity**: In CNNs, each neuron is connected only to a small region of the previous layer, known as the receptive field. This reduces the number of learnable parameters and computational cost, unlike DNNs where neurons are fully connected across layers.
- **Weight Sharing**: CNNs use shared weights within each filter, allowing the same filter to detect patterns across the entire input. This makes them efficient in recognizing local patterns and further reduces the number of parameters.

A typical CNN consists of three main types of layers:

2.3.3.1 Convolutional Layer

The convolutional layer is the core building block of a CNN. It extracts features from the input using a set of learnable filters. Each filter is convolved with the input to produce feature maps. The convolution operation is defined as:

$$z_k^{(l)} = \sum_{c=1}^{C_{l-1}} \sum_{p=1}^{P} \sum_{q=1}^{Q} w_{k,c,p,q}^{(l)} \cdot a_c^{(l-1)} + b_k^{(l)} \tag{2.34}$$

where:

- $z_k^{(l)}$ is the output of the k-th feature map in the l-th layer,
- $a_c^{(l-1)}$ is the activation of the c-th feature map in the $(l-1)$-th layer,
- $w_{k,c,p,q}^{(l)}$ is the weight at position (p,q) in the k-th filter,
- $b_k^{(l)}$ is the bias term.
- P and Q are the spatial dimensions of the filter.

An illustration of the convolution operation between a local receptive field and a filter is shown in Figure 2.3.

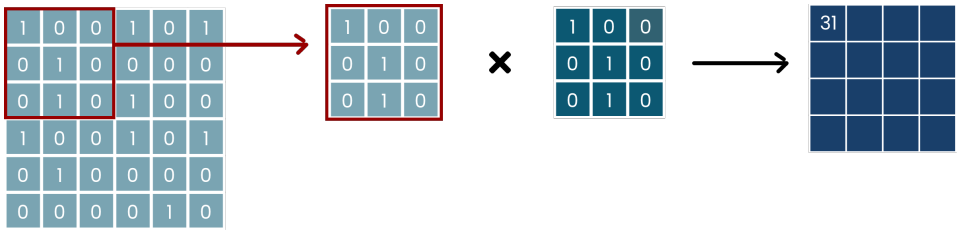

Figure 2.3: Convolution operation between an input (local receptive field) and a filter to produce feature maps.

To introduce non-linearity, the output of the convolutional layer is passed through an activation function, typically the ReLU:

$$f(x) = \max(0, x) \tag{2.35}$$

2.3.3.2 Pooling Layer

The pooling layer reduces the spatial dimensions of the feature maps, simplifying the network. Common pooling methods include Max-pooling, Min-pooling, and Average-pooling. These operations are defined as:

$$\text{Avg-pooling}(x, S) = \frac{1}{hw} \sum_{i,j} (x_{i,j}) \tag{2.36}$$

$$\text{Max-pooling}(x, S) = \max_{i,j} (x_{i,j}) \tag{2.37}$$

where x is the input data, S is a region of size $(h \times w)$, and i, j are indices iterating over the region.

An illustration of the different pooling operations is shown in Figure 2.4.

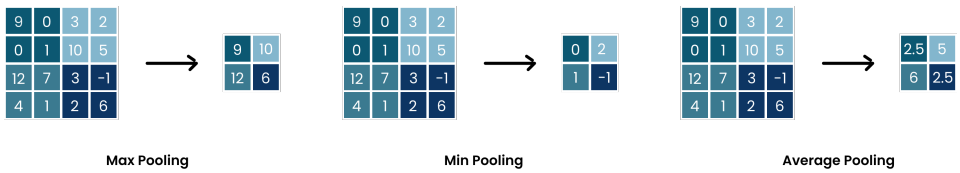

Figure 2.4: Example of pooling operations in CNNs.

2.3.3.3 Fully Connected Layer

After convolutional and pooling layers, the feature maps are flattened into a one-dimensional vector and fed into fully connected layers. These layers apply a non-linear transformation to the input vector, producing the final output. The mathematical formulation is:

$$z^{(l)} = w^{(l)} a^{(l-1)} + b^{(l)} \tag{2.38}$$

where $w^{(l)}$ is the weight matrix, $a^{(l-1)}$ is the output of the previous layer, and $b^{(l)}$ is the bias vector.

A general overview of the CNN architecture, is illustrated in Figure 2.5.

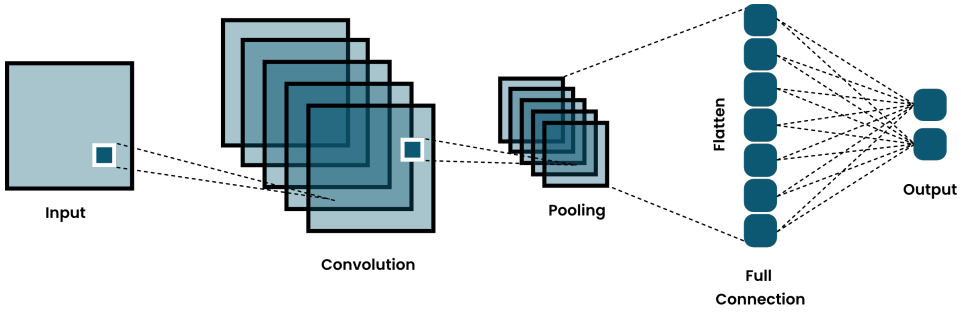

Figure 2.5: General flow of a CNN architecture.

2.3.4 Attention Mechanism

The attention mechanism is a key component in modern neural networks, particularly for tasks involving sequential data such as NLP. It enables models to dynamically focus on the most relevant parts of the input sequence, rather than treating all parts equally. Originally introduced by [41] for machine translation, attention mechanisms have since become a cornerstone of many NLP models. Mathematically, given a sequence of input representations $\mathbf{h} = (h_1, h_2, \ldots, h_n)$, where $h_i \in \mathbb{R}^d$ represents the i-th input feature (e.g., a word embedding), the attention mechanism computes a context vector c as a weighted sum:

$$c = \sum_{i=1}^{n} \alpha_i h_i, \tag{2.39}$$

where α_i is the attention weight assigned to the i-th input feature. These weights are computed using a softmax function:

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)}, \tag{2.40}$$

and e_i is a score that measures the relevance of the i-th input feature to the current context. This score is computed as:

$$e_i = f(h_i, q), \tag{2.41}$$

where q is a query vector representing the current context, and f is a compatibility function that measures the alignment between h_i and q. This alignment determines how much attention the model should pay to the i-th input feature. The compatibility function f can take various forms, such as the dot product or the scaled dot product.

To illustrate how this works in practice, consider the Quranic verse:

وَإِذَا سَأَلَكَ عِبَادِى عَنِّى فَإِنِّى قَرِيبٌ ۖ أُجِيبُ دَعْوَةَ ٱلدَّاعِ إِذَا دَعَانِ ۖ فَلْيَسْتَجِيبُواْ لِى وَلْيُؤْمِنُواْ بِى لَعَلَّهُمْ يَرْشُدُونَ

Translation: And when My servants ask you about Me, I am indeed nearmost. I answer the supplicant's call when he calls upon Me. So let them respond to Me, and let them have faith in Me, so that they may be guided.

This verse carries a **highly positive sentiment**, as it emphasizes Allah's closeness, responsiveness, and benevolence. Words such as أُجِيبُ (I respond), and قَرِيبٌ (near) are central to conveying this sentiment.

To illustrate how attention mechanisms could work, consider how we naturally pay attention when reading this verse. As humans, we instinctively focus on words like أُجِيبُ (I respond) and قَرِيبٌ (near) because they carry the most emotional and contextual weight. These words stand out as key indicators of the verse's meaning and sentiment. Similarly, an attention mechanism in a model could assign higher weights to these words, prioritizing them in its analysis. For instance, the word أُجِيبُ (I respond) might receive the highest attention weight, as it directly conveys Allah's willingness to answer prayers, which is central to the verse's positive sentiment. Likewise, the word قَرِيبٌ (near) could also receive significant weight, as it emphasizes Allah's closeness and accessibility, further reinforcing the positive sentiment of the verse.

In this way, the attention mechanism mirrors how we focus on the most relevant parts of a sentence. By assigning higher weights to key words, the model can prioritize the most important information, enabling it to accurately interpret the sentiment and provide interpretable insights into its decision-making process.

2.3.5 Transformer Architecture

The Transformer architecture, introduced by [150], revolutionized the field of NLP by replacing traditional recurrent and convolutional layers with a mechanism based entirely on **self-attention**. Unlike previous models that processed sequences sequentially, the Transformer processes entire sequences in parallel, making it highly efficient and scalable. The key innovation of the Transformer is its ability to capture long-range dependencies in data without relying on recurrence, which has made it the foundation of many state-of-the-art NLP models, such as Biderctional Encoder Representations from Transformers (BERT) [67], Generative Pre-Trained Transformers (GPT) [131], and T5 [132].

The Transformer consists of an **encoder** and a **decoder**, each composed of a stack of identical layers. The encoder processes the input sequence and generates a set of contextualized representations, while the decoder generates the output sequence by attending to both the encoder's output and its own previous predictions. Each layer in the encoder and decoder includes:

- A multi-head self-attention mechanism,
- A position-wise feedforward network,
- Layer normalization and residual connections.

An overview of the Transformer architecture is illustrated in Figure 2.6.

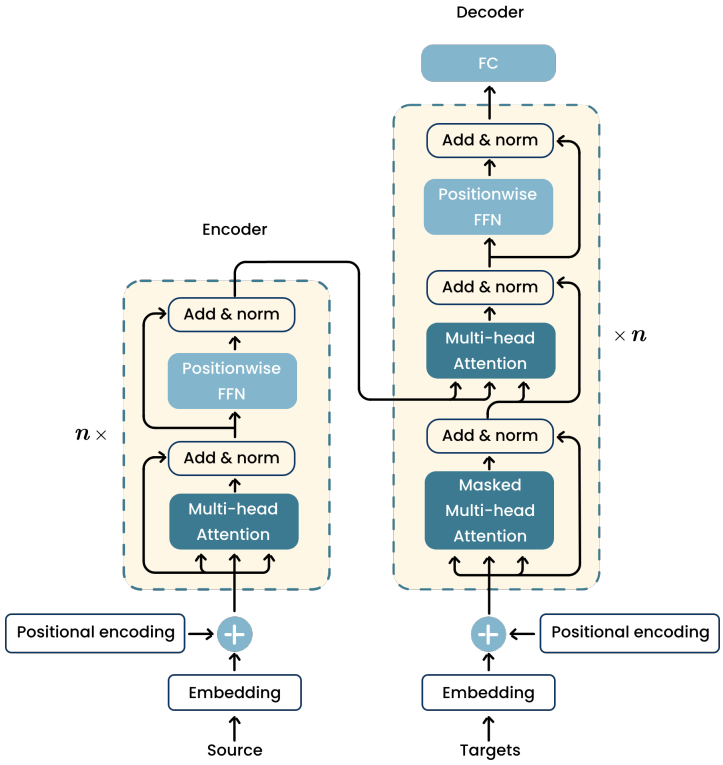

Figure 2.6: The Transformer architecture, consisting of an encoder and a decoder. Each encoder and decoder layer includes multi-head self-attention, a feedforward network, and residual connections.

The **self-attention mechanism** lies at the core of the Transformer. Given an input sequence of word embeddings $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, where $x_i \in \mathbb{R}^d$, the self-attention mechanism projects these embeddings into three learned representations: **queries** (Q), **keys** (K), and **values** (V). The attention scores are computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{2.42}$$

where d_k is the dimensionality of the keys. The softmax function ensures that the attention weights sum to 1, allowing the model to focus on the most relevant parts of the input sequence. This mechanism enables the Transformer to capture contextual relationships between words, regardless of their distance in the sequence.

To enhance the model's ability to capture diverse relationships, the Transformer employs multi-head attention. Instead of computing a single set of attention weights, the model computes multiple attention heads in parallel, each focusing on different parts of the input sequence. The outputs of these heads are concatenated and linearly transformed:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W^O, \tag{2.43}$$

where each head is computed as

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.44}$$

and W_i^Q, W_i^K, W_i^V, and W^O are learned projection matrices. Multi-head attention allows the model to attend to different aspects of the input simultaneously, improving its representational capacity.

Since the Transformer does not process sequences sequentially, it relies on positional encodings to incorporate information about the order of words in the sequence. These encodings are added to the input embeddings and are defined as sinusoidal functions of varying frequencies:

$$\text{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad \text{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right), \tag{2.45}$$

where:

- pos is the position in the sequence,
- i is the dimension index (ranging from 0 to $d/2-1$),
- d is the dimension of the embeddings, and
- 10000 is a user-defined scalar, as proposed by the authors of [150].

The Transformer's ability to process sequences in parallel and capture long-range dependencies has made it the backbone of modern NLP. Its self-attention mechanism eliminates the need for recurrence, enabling faster training and better scalability.

2.3.5.1 Bidirectional Encoder Representations from Transformers

BERT [67] is a Transformer-based model designed exclusively with the encoder component of the Transformer architecture as illustrated in Figure 2.7. Unlike traditional left-to-right or right-to-left language models, BERT employs a bidirectional approach, allowing it to capture rich contextual dependencies by considering both preceding and succeeding words in a sentence.

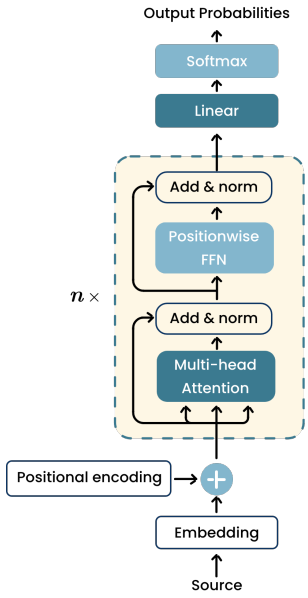

Figure 2.7: The architecture of BERT, consisting of multiple stacked Transformer encoder layers.

BERT is pretrained using two unsupervised tasks:

- **MLM** Instead of predicting the next word in a sequence, BERT randomly masks a percentage of input tokens and trains the model to predict the original token based on its context. This forces BERT to learn bidirectional representations, as illustrated in Figure 2.8.

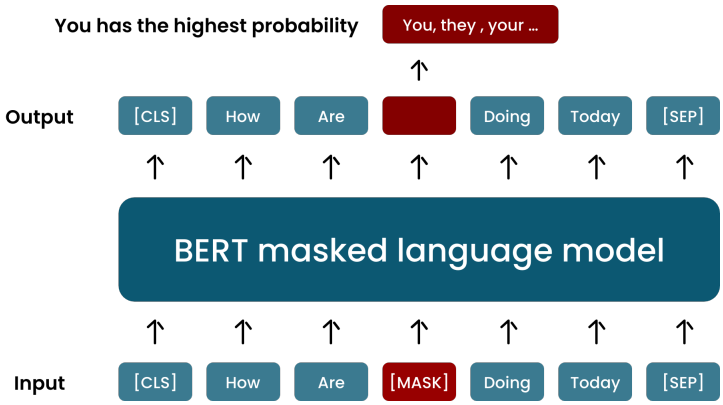

Figure 2.8: Illustration of the MLM used during BERT pretraining.

- **Next Sentence Prediction (NSP)** Given two sentences, BERT is trained to classify whether the second sentence follows the first one in the original text. This task helps the model understand sentence relationships.

The pretrained BERT model can be fine-tuned with minimal task-specific modifications for various NLP applications. Its ability to generate deep contextualized representations has made it a cornerstone of modern NLP advancements.

2.3.6 Mixture of Experts

MoE [95, 98] is a machine learning paradigm that leverages the strengths of multiple specialized models, referred to as experts, to improve overall performance. Each expert is trained to handle a specific subset of the input space, and a gating network determines the contribution of each expert to the final output. This approach allows the model to dynamically adapt to different regions of the input space, making it particularly effective for complex and heterogeneous datasets.

In the MoE framework, the output y for a given input x is computed as a weighted sum of the outputs of the individual experts. Let $E_i(x)$ denote the output of the i-th expert, $G_i(x)$ denote the weight assigned to the i-th expert by the gating network, and n be the number of experts. The final output y is given by:

$$y = \sum_{i=1}^{n} G_i(x) \cdot E_i(x). \tag{2.46}$$

The gating network computes the weights $G_i(x)$ using a softmax function, ensuring that they are non-negative and sum to 1. Specifically, the weights are computed as:

$$G_i(x) = \text{softmax}(x, w_i), \tag{2.47}$$

where w_i represents the parameters of the gating network for the i-th expert. The softmax function ensures that the weights $G_i(x)$ are normalized and can be interpreted as probabilities. This allows the gating network to route inputs to the most relevant experts, effectively partitioning the input space among the experts.

One key advantage of MoE is its ability to perform conditional computation, where only a subset of experts is activated for a given input. This sparsity can significantly reduce computational costs while maintaining high performance. The gating network achieves this by assigning near-zero weights to irrelevant experts, effectively "turning them off" for specific inputs.

Figure 2.9 illustrates the architecture of MoE, where the input is simultaneously provided to both the gating network and the expert networks. The gating network determines the contribution of each expert by assigning weights based on the input. Each expert processes the input independently, and the final output is obtained as a weighted sum of the expert outputs. This

modular design allows MoE to scale efficiently and leverage specialized expert capabilities for diverse tasks.

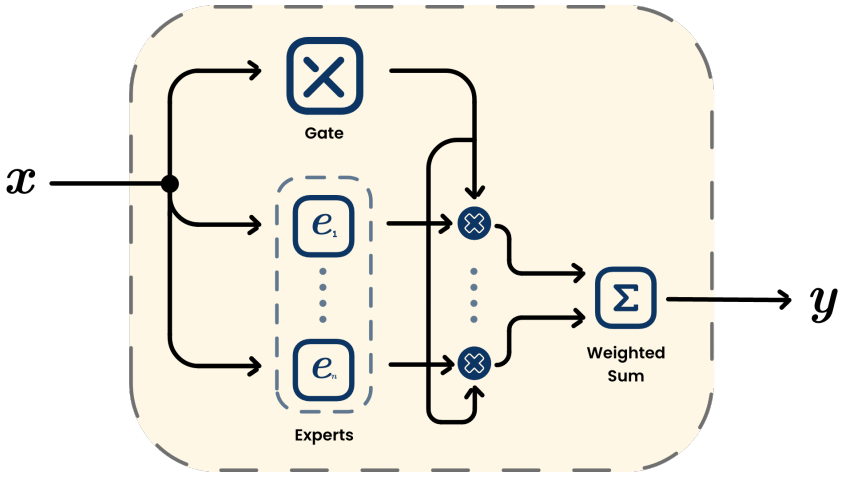

Figure 2.9: Architecture of MoE. The gating network routes inputs to the most relevant experts, and their outputs are combined to produce the final prediction.

2.4 Word Representations

Word representations are fundamental to NLP tasks, as they provide a way to encode textual data in a form that machine learning models can process. Textual data, by its nature, is unstructured and symbolic, consisting of sequences of characters and words that convey meaning through context, grammar, and syntax. However, machine learning algorithms require structured inputs to analyze and learn patterns effectively. Word representations bridge this gap by transforming words, phrases, or entire documents into structured formats that capture their meaning, relationships, or statistical properties.

2.4.1 Classical Representations

Classical word representations are based on statistical and frequency-based methods. These techniques are simple yet effective for many NLP tasks, such as text classification and information retrieval.

2.4.1.1 Bag-of-Words

The Bag-of-Words (BoW) model is one of the simplest word representation techniques in NLP. It represents a document as an unordered collection of words, ignoring grammar, syntax, and word order, but retaining information about word frequencies. This approach treats text as a bag of individual words, where the order of words does not matter, and only their presence and frequency are considered.

Given a vocabulary $V = \{w_1, w_2, \ldots, w_n\}$ of size n, a document d is represented as a numerical vector $\mathbf{v} \in \mathbb{R}^n$. Each element v_i in the vector corresponds to the frequency of word w_i in the document:

$$v_i = \text{count}(w_i, d), \tag{2.48}$$

where $\text{count}(w_i, d)$ is the number of times word w_i appears in document d. For example, if the vocabulary is $V = \{\text{cat}, \text{dog}, \text{run}\}$ and the document d is "the cat runs and the dog runs," the BoW representation would be $\mathbf{v} = [1, 1, 2]$, indicating that "cat" appears once, "dog" appears once, and "run" appears twice.

The BoW model is computationally efficient and easy to implement, making it a popular choice for tasks like text classification and information retrieval. However, it has several significant limitations:

- **Loss of word order:** By disregarding the order of words, BoW fails to capture syntactic structure or relationships between words. For example, the sentences "the cat chased the dog" and "the dog chased the cat" would have the same BoW representation, even though their meanings are different.
- **Semantic ignorance:** BoW does not account for the meaning of words or their relationships. For instance, it treats synonyms (e.g., happy and joyful) as completely unrelated words.
- **High dimensionality:** For large vocabularies, the resulting vectors can be extremely sparse (mostly zeros) and high-dimensional, leading to inefficiencies in storage and computation.

2.4.1.2 Term Frequency-Inverse Document Frequency

Term Frequency-Inverse Document Frequency (TF-IDF) is an extension of the BoW model that weights word frequencies based on their importance in a document relative to a corpus. It aims to highlight words that are more discriminative for a specific document.

The TF-IDF score for a word w_i in a document d from corpus D is computed as:

$$\text{TF-IDF}(w_i, d, D) = \text{TF}(w_i, d) \cdot \text{IDF}(w_i, D), \tag{2.49}$$

where:

- $\text{TF}(w_i, d)$ is the frequency of word w_i in document d, normalized by the total number of words in d. It is computed as:

$$\text{TF}(w_i, d) = \frac{\text{Number of times } w_i \text{ appears in } d}{\text{Total number of words in } d}. \tag{2.50}$$

- IDF(w_i,D) measures how rare or common a word w_i is across the entire corpus D. It is computed as:

$$\text{IDF}(w_i, D) = \log \frac{\text{Total number of documents in the corpus } D}{\text{Number of documents containing } w_i}. \tag{2.51}$$

TF-IDF effectively reduces the weight of common words (e.g., the, and) while increasing the weight of rare but meaningful words. For example, if a word appears frequently in a specific document but rarely in the rest of the corpus, it will have a high TF-IDF score, indicating its importance to that document.

However, TF-IDF still has several limitations:

- **No semantic understanding:** Like BoW, TF-IDF does not capture the meaning or context of words, limiting its ability to model complex linguistic phenomena.
- **Sparsity:** The resulting vectors remain sparse and high-dimensional, especially for large vocabularies.

2.4.1.3 N-grams

N-grams [113] are contiguous sequences of n words or characters extracted from a text. They capture local word order and context, which are lost in the BoW model. By considering sequences of words (or characters), N-grams provide a way to model syntactic patterns and short-range dependencies in text. The value of n determines the type of N-gram:

- **Unigrams ($n = 1$):** Single words or tokens. For example, in the sentence "the cat sat on the mat," the unigrams are:

  $$\{\text{the}, \text{cat}, \text{sat}, \text{on}, \text{the}, \text{mat}\}.$$

  Unigrams are equivalent to the BoW representation and do not capture any word order or context.

- **Bigrams ($n = 2$):** Pairs of adjacent words. For the same sentence, the bigrams are:

  $$\{(\text{the, cat}), (\text{cat, sat}), (\text{sat, on}), (\text{on, the}), (\text{the, mat})\}.$$

  Bigrams capture local word dependencies and are useful for modeling common phrases or collocations (e.g., "New York").

- **Trigrams ($n = 3$):** Triplets of adjacent words. For the same sentence, the trigrams are:

  $$\{(\text{the, cat, sat}), (\text{cat, sat, on}), (\text{sat, on, the}), (\text{on, the, mat})\}.$$

  Trigrams capture slightly longer patterns.

Given a document d, the set of n-grams can be represented as:

$$\text{N-grams}(d,n) = \{(w_i, w_{i+1}, \ldots, w_{i+n-1}) \mid i = 1,2,\ldots,|d|-n+1\}, \tag{2.52}$$

where $|d|$ is the length of the document in words.
However, N-grams suffer from several limitations:

- **Data sparsity:** As n increases, the number of possible N-grams grows exponentially, leading to sparse representations. For example, many possible trigrams or higher-order N-grams may never appear in the training data, making it difficult to generalize.
- **Lack of long-range dependencies**: N-grams only capture local context within a fixed window of n words. They cannot model long-range dependencies or global semantic relationships in text.
- **No semantic understanding:** Like BoW, N-grams do not capture the meaning of words or their relationships.

The quality of word representations directly impacts the performance of downstream NLP tasks. those approaches focuses on capturing surface-level statistics like word frequencies and document-specific importance. While these methods are simple and interpretable, they often fail to capture deeper linguistic properties, such as semantics, syntax, and context.

To address these limitations, more advanced techniques have been developed, such as distributed representations (e.g., word embeddings) and neural language models (e.g., transformers). These methods aim to encode words in a way that reflects their meaning and usage in context, enabling models to generalize better and perform more complex tasks.

2.4.2 Word Embeddings

Word embeddings are dense, low-dimensional vector representations of words that capture semantic and syntactic relationships. Unlike classical representations like BoW or TF-IDF, which rely on sparse, high-dimensional vectors based on word frequencies, word embeddings encode meaning in a continuous vector space. In this space, words with similar meanings are located close to each other, enabling models to generalize better and capture nuanced relationships between words. For example, in a well-trained embedding space, the vectors for "king" and "queen" will be close to each other, as will the vectors for "man" and "woman," reflecting their semantic similarity.

To illustrate, consider the following analogy:

$$\text{king} - \text{man} + \text{woman} \approx \text{queen}.$$

This means that the vector difference between "king" and "man" is similar to the vector difference between "queen" and "woman." Such relationships are learned automatically by word embedding models, making them powerful tools for capturing linguistic patterns.

The most widely used word embedding methods include Word2Vec, GloVe, and FastText. Each of these methods learns embeddings by analyzing large corpora of text, but they differ in their approaches and the types of relationships they capture.

2.4.2.1 Word2Vec

Word2Vec, introduced by Mikolov et al. [122], is a pioneering word embedding technique that learns word representations by predicting words based on their context. It employs two architectures (Figure 2.10):

- **Continuous Bag-of-Words (CBoW)** Predicts a target word based on its surrounding context words. For example, given the context words "the cat sat on the," CBoW predicts the target word "mat."
- **Skip-gram**: Predicts context words given a target word. For example, given the target word "sat," Skip-gram predicts the surrounding context words "the," "cat," "on," and "the."

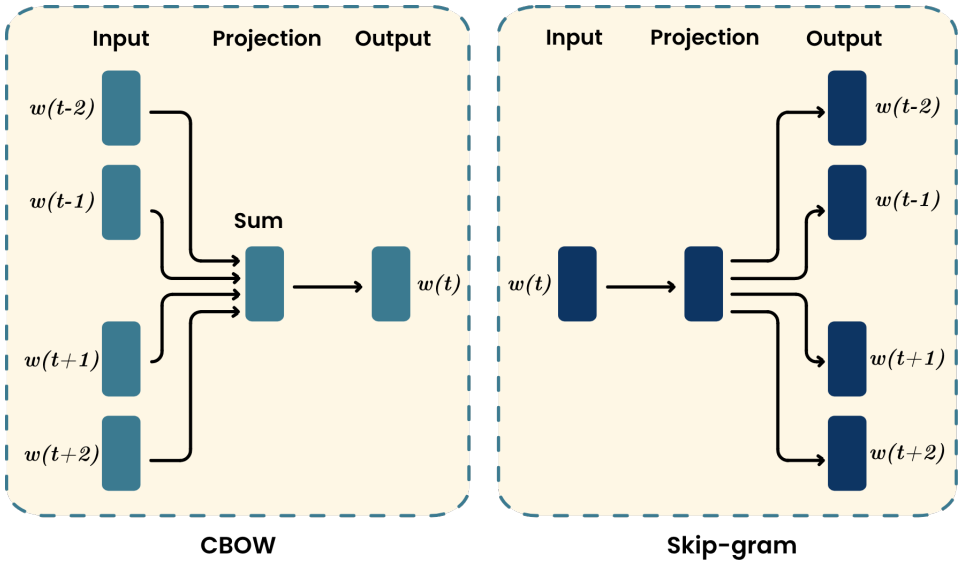

Figure 2.10: The two architectures of Word2Vec: (a) CBOW predicts a target word from context words, and (b) Skip-gram predicts context words from a target word.

Word2Vec embeddings are learned by training a shallow neural network on a large corpus of text. The model maps words to dense, low-dimensional vectors in a continuous space, where words with similar meanings are located close to each other.

One of the key strengths of Word2Vec is its computational efficiency. The model uses techniques like hierarchical softmax or negative sampling to reduce the computational cost of training, making it scalable to large corpora.

However, Word2Vec has several limitations:

- **Fixed representations**: Each word is assigned a single vector, regardless of its context. This means that polysemous words (words with multiple meanings) are represented by the same vector, which can lead to ambiguity. For example, the word "bat" would have the same vector in the sentences "The bat flew through the cave" (referring to the animal) and "He swung the bat to hit the ball" (referring to the sports equipment). Word2Vec cannot distinguish between these meanings, as it generates a single representation for each word which limits the model's ability to handle words with multiple meanings.
- **Out-of-vocabulary words**: Word2Vec cannot generate embeddings for words not seen during training, limiting its ability to handle rare or new words.

2.4.2.2 GloVe

Global Vectors for Word Representation (GloVe), introduced by Pennington et al. [130], is another widely used word embedding method. Unlike Word2Vec, which learns embeddings based on local context (e.g., predicting neighboring words), GloVe leverages global word co-occurrence statistics from the entire corpus. It constructs a word-context matrix X, where each entry X_{ij} represents the number of times word j appears in the context of word i. GloVe then factorizes this matrix to obtain word embeddings that capture the relationships between words.

The key idea behind GloVe is that the ratio of co-occurrence probabilities between words can encode meaningful semantic relationships. Specifically, GloVe aims to learn word vectors $\mathbf{w}_i$ and context vectors $\tilde{\mathbf{w}}_j$ such that their dot product approximates the logarithm of the co-occurrence probability:

$$\mathbf{w}_i^\top \tilde{\mathbf{w}}_j + b_i + \tilde{b}_j \approx \log(X_{ij}) \tag{2.53}$$

where:

- $\mathbf{w}_i$ and $\tilde{\mathbf{w}}_j$ are the word and context vectors for words i and j, respectively,
- b_i and $\tilde{b}_j$ are bias terms for words i and j,
- X_{ij} is the co-occurrence count of word i and context word j.

GloVe embeddings are trained by optimizing the following objective function:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij}) \left(\mathbf{w}_i^\top \tilde{\mathbf{w}}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2 \tag{2.54}$$

where $f(X_{ij})$ is a weighting function that reduces the influence of rare co-occurrences. The weighting function is defined as:

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{x_{\max}}\right)^{\alpha} & \text{if } X_{ij} < x_{\max}, \\ 1 & \text{otherwise,} \end{cases} \tag{2.55}$$

where $x_{\max}$ is a cutoff value (typically 100) and α is a hyperparameter (typically 0.75). The weighting function in GloVe ensures that both frequent and rare co-occurrences contribute meaningfully to training by downweighting overly frequent words while ensuring that rare words are not completely ignored.

However, GloVe has several limitations:

- **Fixed representations**: Like Word2Vec, GloVe generates a single vector for each word, regardless of its context.
- **Dependence on co-occurrence statistics**: GloVe relies heavily on the quality and coverage of the co-occurrence matrix. Rare words or words with sparse co-occurrence patterns may not be well-represented in the embedding space.

2.4.2.3 FastText

FastText, developed by Facebook AI Research [52], extends Word2Vec by representing words as bags of character n-grams. This approach allows FastText to generate embeddings for Out Of Vocabulary (OOV) words by breaking them into subword units. For example, the word "running" can be represented as the sum of its character n-grams: "run", "runn", "unning", etc. This subword information enables FastText to capture morphological patterns and handle rare or unseen words effectively.

The idea behind FastText is that a word's embedding is the sum of the embeddings of its constituent n-grams. Given a word w and its set of character n-grams $\mathcal{G}(w)$, the word embedding $\mathbf{v}_w$ is computed as:

$$\mathbf{v}_w = \sum_{g \in \mathcal{G}(w)} \mathbf{v}_g \tag{2.56}$$

where $\mathbf{v}_g$ is the embedding of the n-gram g. For example, the word "running" might be represented as:

$$\mathbf{v}_{\text{running}} = \mathbf{v}_{\text{run}} + \mathbf{v}_{\text{runn}} + \mathbf{v}_{\text{unning}}. \tag{2.57}$$

FastText uses the same architectures as Word2VecCBoW and Skip-grambut applies them to n-grams instead of whole words. This allows FastText to:

- **Handle out-of-vocabulary words**: By breaking words into subword units, FastText can generate embeddings for words not seen during training.
- **Capture morphological patterns**: FastText is particularly effective for morphologically rich languages (e.g., Turkish) where words can have many inflected forms.

However, FastText has several limitations:

- **Fixed representations**: Like Word2Vec and GloVe, FastText generates a single vector for each word, regardless of its context. This means that polysemous words are represented by the same vector.
- **Increased complexity**: The use of subword units increases the complexity of the model, as it requires storing and processing embeddings for a large number of n-grams.

2.4.3 Contextualized Embeddings

Contextualized embeddings address the limitations of traditional word embeddings by generating dynamic representations of words based on their context. Unlike Word2Vec, GloVe, or FastText, which assign a fixed vector to each word, contextualized embeddings produce different representations for the same word depending on its usage in a sentence. This enables them to capture polysemy and nuanced linguistic phenomena.

The development of transformer architectures, as discussed in Section 2.3.5, has been instrumental in advancing contextualized embeddings. Transformer-based models like BERT (Section 2.3.5.1) and GPT generate context-aware representations by leveraging self-attention mechanisms. These models have revolutionized NLP by achieving state-of-the-art performance on a wide range of tasks, including question answering, text classification, and machine translation.

Contextualized embeddings have become the foundation of modern NLP, enabling machines to understand and generate human language with unprecedented accuracy and fluency. Their ability to capture context and meaning has made them indispensable tools for a wide range of applications.

2.5 Knowledge Sharing Paradigms in NLP

The limitations of supervised learning in low-resource scenarios, particularly for dialectal Arabic, necessitate learning paradigms that promote knowledge reuse and generalization. Among the most effective solutions are transfer learning and multitask learning, both of which address data scarcity by leveraging shared linguistic patterns across tasks or domains. These paradigms move beyond task-specific modeling by enabling the transfer of representations learned from resource-rich settings, such as MSA, to improve performance on under-resourced dialects. This section presents an overview of these two learning frameworks.

2.5.1 Transfer Learning

Transfer Learning [129] is a paradigm that enables the reuse of knowledge learned from one task or domain to enhance performance on a different, but related, task or domain. Traditionally, machine learning systems are trained from scratch for each specific task, requiring large annotated datasets. This poses a significant challenge for low-resource settings, where labeled data is scarce. Transfer learning mitigates this by leveraging prior learning, allowing models trained on data-rich tasks to be adapted to low-resource problems with minimal additional supervision.

In NLP, transfer learning has been widely adopted following the success of pretrained language models such as BERT. These models are trained on massive corpora using self-supervised objectives (e.g., MLM or Next Sentence Prediction) and are later fine-tuned on specific downstream tasks like sentiment analysis or question answering. By pretraining on general-purpose text, the models capture rich syntactic and semantic knowledge that is transferable across tasks and domains.

A particularly impactful application of transfer learning is cross-lingual transfer, where knowledge from a high-resource language (e.g., English or MSA) is transferred to improve performance on a low-resource language or dialect (e.g., Algerian Arabic). This is especially effective when the source and target languages share typological or structural similarities. In this thesis, transfer learning is a foundational mechanism for bridging the resource gap between MSA and Arabic dialects. It enables the adaptation of MSA-pretrained embeddings to dialect-specific tasks such as dialect identification and sentiment analysis, significantly improving performance without requiring large amounts of dialectal training data.

2.5.2 Multitask Learning

Multitask Learning [60] is a learning paradigm in which a single model is trained to perform multiple tasks simultaneously. Unlike traditional single-task learning where each task requires a separate model, Multitask Learning enables parameter sharing across tasks, encouraging the model to learn a more generalized and robust representation. The central idea is that learning tasks in parallel acts as a guide to the model to capture underlying structures that benefit all tasks involved.

Multitask Learning is beneficial in scenarios where one task has abundant training data while the other suffers from data scarcity. In such cases, the auxiliary task with richer data helps regularize the learning process and provides useful features that improve performance on the primary task. The key to successful multitask learning lies in selecting related tasks and designing shared architectures that allow for effective information exchange while preserving task-specific nuances.

2.6 Conclusion

This chapter has provided a comprehensive overview of the challenges and techniques involved in processing Arabic and its dialects using NLP. Arabic's linguistic complexity, from its root-and-pattern morphology to its syntactic flexibility and phonological diversity, demands sophisticated computational models capable of handling its intricacies. The Algerian dialect, with its rich history and linguistic diversity, exemplifies the challenges faced by low-resource languages in NLP.

The chapter also explored traditional machine learning approaches, such as Decision Trees, Naïve Bayes, and Support Vector Machines, which have laid the groundwork for many NLP tasks. These methods, while effective for certain applications, are often limited in their ability to capture the nuanced relationships and contextual dependencies present in language. The advent of deep learning and neural networks, particularly RNNs, LSTMs, GRUs, CNNs, and Transformers, has revolutionized the field, enabling more accurate and context-aware language processing. These models have significantly improved the ability of NLP systems to handle sequential data, capture long-range dependencies, and generate dynamic representations of words.

Subsequently, the chapter discussed word representations, from classical methods like BoW and TF-IDF to modern word embeddings such as Word2Vec, GloVe, and FastText. It also explored contextualized embeddings, which have become the cornerstone of modern NLP, enabling models to generate dynamic representations of words based on their context.

Finally, the chapter introduced two critical learning paradigms, transfer and multitask learning, that are particularly effective for addressing the limitations of low-resource language processing. These paradigms are not only theoretically significant but also form the backbone of the methodologies proposed in this thesis.

By integrating these techniques and learning paradigms, this chapter establishes the theoretical foundation for the subsequent research. Together, they offer scalable and adaptable solutions for advancing NLP in complex and under-resourced linguistic settings.

CHAPTER 3

LITERATURE REVIEW

This chapter provides a critical review of existing work related to Arabic dialect identification and sentiment analysis. By examining traditional, deep learning, and transformer-based approaches, it highlights the current limitations and research gaps in low-resource Arabic NLP. These insights serve as a guide for formulating the methodologies and innovations proposed in the following chapters.

As demonstrated in the previous chapter, ML and Deep Learning (DL) have emerged as transformative tools for tackling complex linguistic challenges. Among these, ADI and Sentiment Analysis stand out as two particularly intricate problems, each with its own unique set of challenges. Arabic Dialect Identification is particularly demanding because it requires the model to learn fine-grained distinguishing features that separate closely related dialects, while ADI is hindered by high lexical overlap, rich morphological diversity, and subtle contextual variations among dialects. These factors make it difficult to distinguish between them, especially when they share significant vocabulary with MSA and each other. Conversely, sentiment analysis in Arabic dialects requires the model to capture the internal semantic and contextual nuances of a given dialect, and it struggles with limited annotated datasets, the presence of code-switching, and the nuanced expression of sentiment that varies across dialects and cultural contexts.

By focusing on these two tasks, this review provides a structured foundation for examining key dimensions of dialectal Arabic processing. Dialect identification reflects cross-dialectal distinctions, requiring systems to capture subtle linguistic differences across closely related varieties, often blurred by shared vocabulary and overlapping morphological features. In contrast, sentiment analysis targets intra-dialectal nuance, where the challenge lies in detecting emotional tone and semantic intent expressed through informal, culturally embedded, and often non-standard language. This dual focus not only enables a broader exploration of methods suitable

for handling different levels of language variation, but also offers insight into how transferable linguistic knowledge may be adapted to serve low-resource dialectal contexts. Moreover, these tasks together serve as a meaningful basis for reviewing the capacities and limitations of existing approaches in the field.

This chapter provides a comprehensive review of state-of-the-art approaches in both ADI and sentiment analysis, categorizing them into three major paradigms: Traditional Machine Learning, Deep Learning (Non-Transformer), and Transformer-Based (Transfer Learning) methods. By critically analyzing representative works in each category, we highlight their methodologies, strengths, limitations, and the evolution of the field. The first part of focuses on ADI, reviewing the progression from traditional feature-based methods to advanced transformer-based models. The second part shifts to sentiment analysis, exploring how similar methodologies have been adapted to address the unique challenges of sentiment analysis in Arabic dialects. Together, this review contextualizes the contributions of this thesis and highlights open challenges and future research directions in both areas.

3.1 Related Work in Dialect Identification

ADI seeks to classify text into dialect-specific categories despite extensive vocabulary sharing with MSA and among dialects themselves. Over the past decade, researchers have tackled this task through three methodological paradigms mentioned earlier. In the following sections, we discuss each in detail, emphasizing key contributions, challenges, and the field's evolution.

3.1.1 Traditional Machine Learning Approaches

Traditional methods for ADI rely on manual feature engineering combined with probabilistic or kernel-based classifiers. While these approaches are computationally efficient and more interpretable, they are limited in capturing complex, context-dependent features as discussed in section 2.4.1. Nonetheless, many shared tasks and studies have shown that when paired with well-selected features, statistical classifiers remain effective.

In the Nuanced Arabic Dialect Identification (NADI) 2020 shared task [9], Aliwy et al. [23] explored ADI across 21 countries using statistical classifiers such as Naïve Bayes, Logistic Regression, and Decision Trees. Their approach leveraged word and character n-grams as features, with additional clustering techniques applied to mitigate noise from MSA tweets in the training data. The combination of these classifiers through a voting mechanism, along with clustering, resulted in an F1-score of 20.05% on the test set.

In the NADI 2021 shared task [10], Ali et al. [126] also used a machine learning-based approach for ADI. Their methodology involved extracting features using TF-IDF, and employing classifiers such as Complement Naïve Bayes (CNB), SVM, Decision Tree, Logistic Regression, and Random Forest. Among these, the CNB classifier achieved the best performance, with F1-scores

of 12.99% and 18.72% for MSA-country level (Subtask 1.1) and DA-country level (Subtask 1.2), respectively, and 3.51% and 4.55% for MSA-province level (Subtask 2.1) and DA-province level (Subtask 2.2). The corresponding accuracies were 23.24%, 37.16%, 3.38%, and 4.8%. These results underscore the challenge of fine-grained dialect identification.

The trend of leveraging statistical classifiers continued in the NADI 2022 shared task [11], where the SUKI team [96] employed a Naïve Bayes classifier trained on character n-grams (ranging from one to four). Their model achieved an F1-score of 19.63% on Test-A and 10.58% on Test-B.

Beyond the NADI shared tasks, Mishra et al. [123] explored a broader range of ML and DL models on the Multi-Arabic Dialect Applications and Resources (MADAR) dataset [53], utilizing various feature extraction techniques, including n-grams (word and character-level) and language model probabilities. They experimented with several classifiers, including Multinomial Naïve Bayes (MNB), SVM, and DL models, to assess the impact of these features on classification performance. Their findings highlighted the effectiveness of traditional ML classifiers over neural networks. Their best performing model aMNB classifier using word and character-level 5-grams alongside language model probabilities, achieved an accuracy of 66.31% and an F1 score of 66.21%. Additionally, an SVM classifier incorporating n-grams and dialect probabilities, obtained 67.20% accuracy and a 57.90% F1-score in subtask 2.

Similarly, Talafha et al. [148] also explored the MADAR shared task by employeing TF-IDF representations and found that a Linear Support Vector Classifier (LinearSVC) performed competitively when provided with curated features. They further enhanced classification by introducing a dialect weighting scheme through user voting, where predictions were adjusted to give more emphasis to less frequent dialects, this was achieved by assigning higher weights to underrepresented dialects and modifying the impact of specific tweet types (e.g., increasing the weight of retweets and reducing the influence of unavailable tweets), which led to 76.20% accuracy and a 69.86% F1-score on the test set. By reweighting underrepresented dialects, they improved over Mishra et al.'s [123] subtask 2 results by 11.96% in F1-score, this improvement underscores the importance of addressing class imbalance.

Using the same data, Baimukan et al. [42], introduced a multi-classifier approach, using classifiers trained at the city, country, and region levels. These classifiers were trained on the Madar dataset with n-gram features. Instead of standard 5-gram language model scores, the authors employed scores from aggregated models at each hierarchical level to support the main classifier MNB, which focused on the 26 MADAR labels. The aggregated country classifier improved accuracy by 0.79% (to 77.12%) and F1 score by 0.73% (to 74.83%). However, combining classifiers across different levels introduced noise, limiting further gainsdespite outperforming Talafha et al. [148] by 4.97%.

Lastly, El-Haj et al. [70] introduced a novel Subtractive Bivalency Profiling (SBP) approach to enhance ADI by capturing grammatical variations. Their study focuses on distinguishing between

Egyptian, North African, Gulf, Levantine dialects, and MSA, tackling challenges such as bivalency and code-switching. Their SVM-based model achieved over 76% accuracy and maintained 66% performance on unseen data.Despite its promise, the approach is sensitive to feature selection and computational constraints, which limits its scalability.

Overall, these studies demonstrate the enduring relevance of traditional machine learning approaches for ADI, particularly when combined with well-designed feature engineering techniques. Even in the era of deep learning, statistical classifiers remain competitive, especially in scenarios with limited training data, as they typically require less data to achieve reasonable performance compared to deep learning models.

3.1.2 Deep Learning Approaches (Non-Transformer)

DL has revolutionized ADI by automating the feature extraction process, enabling models to learn complex representations directly from raw data. This section reviews neural network architectures that do not leverage transformer-based pre-training.

In their work, Issa et al. [93] explored country-level ADI using the NADI 2021 dataset. They compared two model, an LSTM with pretrained word embeddings, and a feature-augmented LSTM incorporating engineered linguistic features. Their results showed that the LSTM with pretrained embeddings outperformed the feature-augmented version, achieving 41.36% accuracy and an F1-score of 22.10% on the development set.

In the NADI 2022 shared task, AlShenaifi et al. [31] evaluated both machine learning models like Logistic Regression and SVM, and deep learning models including BiLSTM and AraBERT. Their results identified BiLSTM as the best-performing model, achieving 39.9% accuracy and 22.4% F1-score on the Test-A, and 23.7% accuracy with a 9.3% F1-score on the Test-B. This represents a notable improvement over purely traditional approaches, such as the SUKI team's Naïve Bayes classifier [96], which relied on character n-grams (F1-scores of 19.63% on Test-A and 10.58% on Test-B). While both teams tackled the same dataset and task, BiLSTM demonstrated the superiority of sequence-aware architectures over static n-gram features, improving the F1-score by 2.77% on Test-A. However, on Test-B, the Naïve Bayes classifier outperformed the BiLSTM, achieving a 1.28% higher F1-score. This mixed performance highlights the strengths of deep learning models in capturing sequential dependencies, as seen in Test-A, while also underscoring the challenges of generalizing to real-world dialectal data, particularly in Test-B, where both models struggled to achieve an F1-score above 11%.

De Francony et al. [66] proposed two approaches for ADI using the MADAR dataset, a hierarchical DL model and a vote-based probabilistic classifier. The hierarchical model employed a two-stage classification process, first predicting a broader dialect group using a BiLSTM, followed by a second-stage model that classified the specific dialect within that group. The vote-based approach combined an MNB classifier and a Random Forest classifier, using hard voting to select the final prediction, their results showed that the vote-based classifier outperformed the

hierarchical deep learning approach, achieving the highest F1-score of 63.02%. However, this approach underperformed Mishra et al.'s MNB classifier (66.21% F1) [123] and Baimukan et al.'s hierarchical multi-classifier framework [42] (74.83% F1). This disparity highlights challenges faced by De Francony et al.'s BiLSTM architecture in capturing robust representations from the sparse, fine-grained MADAR corpus. Mishra et al. prioritized explicit feature engineering to encode linguistic patterns and dialectal markers, which proved effective in this low-resource scenario. Similarly, Baimukan et al. reduced ambiguity through structured hierarchical n-gram aggregation across geographic levels. By contrast, the BiLSTM's sequential dependency modeling struggled with dialectal complexity and limited training data, suggesting that this specific implementation may have been constrained by its architectural assumptions or optimization dynamics.

Similarly, in their study, Fares et al. [78] addressed the challenge of ADI using the MADAR Corpora. They proposed multiple models combining frequency-based features and deep learning. It uses character and word-level TF-IDF features as input to a neural network with two dense layers. Simultaneously, an MNB model processes the same features for statistical classification. The outputs from both models are combined using log probability averaging, this approach achieved an F1-score of 65.35% and an accuracy of 65.75% on the test set. While also De Francony et al. [66] leveraged ensemble techniques to improve performance, Fares et al.'s hybrid approach outperformed De Francony et al.'s vote-based classifier, achieving a higher F1-score (65.35% vs. 63.02%). This suggests that combining deep learning with statistical methods may be more effective. However, Mishra et al.'s MNB classifier (66.21% F1) [123] and Baimukan et al.'s hierarchical multi-classifier framework (74.83% F1) [42] achieved stronger results, indicating that explicit feature engineering and hierarchical aggregation strategies remain competitive in low-resource settings.

Non-transformer DL approaches show limited but promising results in ADI. While architectures like BiLSTM and hybrid frameworks outperform basic statistical models in capturing sequential or combined features, they struggle to match the performance of feature-engineered (e.g., Mishra et al. [123]) or hierarchically structured (e.g., Baimukan et al. [42]) traditional methods. This underscores the challenge of learning robust dialect representations from sparse data without explicit linguistic priors. Pretrained embeddings (e.g., Issa et al. [93]) partially bridge this gap, suggesting that integrating domain knowledge with neural architectures could mitigate data scarcitya direction further advanced by transformer-based methods, as discussed next.

3.1.3 Transformer-Based Approaches (Transfer Learning)

The emergence of transformer architectures and pre-trained language models has dramatically advanced ADI. These models are pre-trained on large corpora, capturing deep contextual and semantic nuances, and then fine-tuned on dialect-specific data.

El Mekki et al. [72] proposed a BERT-based Multi-Task Learning model for Arabic dialect and MSA identification at both country and province levels using NADI 2021. Their model leverages MARBERT, combined with task-specific attention layers to extract discriminative features for each task. The model achieved F1-scores of 21.4%, 30.64%, 5.35%, and 7.32% with corresponding accuracies of 33.84%, 50.30%, 5.72%, and 7.92% for Subtasks 1.1, 1.2, 2.1, and 2.2, respectively. In contrast, traditional ML approaches, such as those by Ali et al. [126], relied on TF-IDF features and classifiers like CNB, achieving lower F1-scores and accuracies for both country and province levels. Reflecting the limitations of traditional methods in handling fine-grained dialect identification. This comparison highlights the superiority of transformer-based models in capturing complex, context-dependent features. However, the province-level classification remains challenging across both approaches, with F1-scores consistently below 10%, reflecting the inherent difficulty of fine-grained dialect identification.

Similarly, AlKhamissi et al. [24] tackled the NADI 2021 shared task by fine-tuning MARBERT with an adapter-based approach. Instead of modifying all layers, they inserted lightweight adapter modules into each transformer layer, allowing only these layers to be updated while keeping the pre-trained MARBERT parameters frozen. Additionally, they introduced a novel Vertical Attention mechanism that attends across transformer layers, enabling deeper feature extraction. Their final model was an ensemble combining four configurations, fully fine-tuned MARBERT, adapter-tuned MARBERT, fine-tuned MARBERT with Vertical Attention, and a fine-tuned model with a linear learning rate schedule. This ensemble achieved on the test set the F1-scores of 22.38%, 32.26%, 6.43%, and 8.60% with corresponding accuracies of 35.72%, 51.66%, 6.66%, and 9.46% for Subtasks 1.1, 1.2, 2.1, and 2.2, respectively. While both this study and El Mekki et al.'s [72], leverage MARBERT and transformer-based approaches, this approach outperformed El Mekki et al.'s [72] model across all subtasks, achieving higher F1-scores and accuracies. This highlights the effectiveness of adapter-based fine-tuning and Vertical Attention in enhancing transformer-based models for ADI. However, again both studies underscore the persistent challenges of province-level classification, where F1-scores remain below 10%, reflecting the inherent difficulty of the task.

Humayun et al. [90] proposed an ensemble-based approach for ADI also using the NADI 2021 dataset. Their approach integrates three individual classifiers, with the final prediction obtained by averaging their outputs. The first classifier fine-tunes the MARBERT model by training only selected intermediate layers while keeping the rest of the transformer frozen. The second classifier employs the adapter-based fine-tuning mechanism. The third classifier is a feature-based model that utilizes concatenated hidden-layer representations from a pre-trained MARBERT without fine-tuning. The ensemble achieved an overall accuracy of 53.96% on the country-level classification development set. By diversifying fine-tuning strategies freezing layers versus updating adapters the authors achieved better generalization, though computational costs increased.

AlHassan et al. [40] using the NADI 2022 dataset, proposed two main approaches, a multi-task learning model which leverages a shared AraBERT encoder, and three task-specific classification heads for predicting region, area, and country-level dialects. The multi-task model achieved an F1-score of 32.63% on Test-A and 15.61% on Test-B for ADI subtask. This represents a significant improvement over earlier approaches, such as the BiLSTM model by AlShenaifi et al. [31], which achieved an F1-score of 22.4% on Test-A and 9.3% on Test-B. The 10.23% increase in F1-score on Test-A and 6.31% increase on Test-B highlight the effectiveness of multi-task learning and transformer based approaches in capturing shared linguistic features across different levels of dialect granularity. However, both studies underscore the persistent challenges of province-level classification, particularly in Test-B, where performance remains relatively low.

Building on these advancements, Bayrak et al. [47] also tackled the NADI 2022 shared task. They employed domain-adapted BERT-based models with language-specific preprocessing to improve performance. Their best-performing models were MARBERTv2, achieving an F1-score of 33.89% and an accuracy of 51.66% on Test-A, and MARBERT which achieved an F1-score of 31.66% and an accuracy of 49.18% on Test-A. On Test-B, MARBERTv2 scored 17.19% F1 and 34.87% accuracy, while MARBERT scored 17.51% F1 and 35.14% accuracy. Compared to AlHassan et al.'s [40] multi-task model, Bayrak et al.'s MARBERTv2 achieved a higher F1-score on both Test-A and Test-B, demonstrating the effectiveness of domain-adapted BERT models and language-specific preprocessing. However, both approaches struggled to achieve high F1-scores, especially in Test-B, reflecting the persistent challenges of fine-grained ADI.

Shammary et al. [138] in the NADI 2022 Shared Task (ADI), explored three main approaches. The first was a traditional TF-IDF-based model. The second approach involved fully fine-tuning MARBERT. The third approach combined MARBERT with adapters that are inserted between transformer layers, allowing for efficient adaptation without updating the full model, they also used data augmentation techniques to enhance generalization. The fully fine-tuned MARBERT model achieved an F1-score 18.62% and 16.68% on TEST-A and TEST-B, respectively, with corresponding accuracies of 32.18% and 33.38%. The adapter-based MARBERT model with data augmentation performed slightly better on TEST-B, achieving an F1-score of 17.67 and accuracy of 33.92%, despite lower results on TEST-A (F1-score: 4.85%, accuracy: 11.52%). Compared to Bayrak et al.'s MARBERTv2 [47], the fully fine-tuned MARBERT achieved significantly lower F1-scores (18.62% vs. 34.87% on Test-A and 16.68% vs. 17.19% on Test-B). However, their adapter-based MARBERT demonstrated competitive performance on Test-B (17.67% F1-score vs. 17.19%), highlighting the potential of adapter-based fine-tuning and data augmentation for improving generalization. Despite these advancements, both studies underscore the challenges of province-level classification, particularly in Test-B, where F1-scores remain below 20%.

Nwesri et al. [127] participated in the NADI 2023 Shared Task [7], focusing on country-level ADI. They fine-tuned the MARBERTv2 transformer on a modified training dataset, where tweet length was increased by combining every two adjacent tweets from the same dialect into a single,

longer tweet. This data augmentation provided richer contextual information, enabling the model to better capture dialectal patterns. Their approach achieved an F1 score of 82.87% and an accuracy 82.86%.

Building on this, Adel et al. [14] proposed a voting ensemble approach for ADI in the NADI 2023 Shared Task, combining predictions from multiple fine-tuned models. The ensemble included a MARBERT model fine-tuned with a standard approach and an enhanced variant where averaged intermediate layers were concatenated with the final layers, followed by convolutional filters and max-pooling. This combination of fine-tuning, layer concatenation, and convolutional enhancements, coupled with ensemble voting, achieved an F1 score of 83.73%, and an accuracy of 83.67%. The improvement over Nwesri et al.'s [127] results can be attributed to the ensemble strategy, which leverages the strengths of multiple models, and the convolutional enhancements, which capture local patterns in the text more effectively than tweet concatenation alone.

Further advancing these efforts, Elkaref et al. [75] in the NADI 2023 Shared Task on ADI at the country level. Their approach named NLPeople integrated several strategies: employing MARBERTv2, enhancing performance through context augmentation by retrieving and appending similar texts from the training data for richer input context, implementing a staged fine-tuning process, first training on older task data from NADI 2020 and 2021 before final tuning on the 2023 dataset, and model ensembling combines predictions from multiple models via score stacking or majority voting to improve robustnes. This methodology led to a macro-averaged F1 score of 87.27% and an accuracy of 87.22%, surpassing both Nwesri et al. [127] and Adel et al. [14] by a significant margin. The superior performance of Elkaref et al.'s approach can be attributed to the staged fine-tuning process, which allows the model to learn general dialectal patterns from older datasets before specializing on the 2023 data, and the context augmentation, which provides richer input context by incorporating similar texts. Additionally, their ensembling strategy further refines predictions by combining multiple models. This progression highlights the effectiveness of data augmentation, ensemble techniques, and staged fine-tuning in improving ADI performance. While Nwesri et al. [127] demonstrated the value of contextual enrichment through tweet concatenation, Adel et al. [14] showed the benefits of layer concatenation and convolutional enhancements. NLPeople system combined these strategies with staged fine-tuning and ensembling, achieving the highest performance and setting a new benchmark for ADI in the NADI 2023 Shared Task.

In the NADI 2024 Shared Task [8], Kanjirangat et al. [102] tackled Multi-label Arabic Dialect Identification (MLDID), where input samples could belong to multiple dialect classes. The best approach involves an unsupervised cross-encoder with post-filtering. The task is modeled as a retrieval problem, where the training data is treated as a labeled database, and each test sample is treated as a query. MARBERTv2 is used to compute sentence embeddings for the training data. A semantic search is performed to retrieve the top K=10 most similar training samples for each query. The labels of these retrieved samples are then used to assign labels to the query. To refine

the predictions, a post-filtering step is applied, where only the top N=8 multi-label predictions are retained. This filtering step was found to maximize the F1-score on the development set. This approach maximized the F1-score on the development set and achieved the best results, with an F1 score of 43.27% and an accuracy of 71.88% on the test set.

In the same shared task on MLDID, Karoui et al. [103] proposed an ensemble of fine-tuned BERT-based models enhanced by a Similarity-Induced Mono-to-Multi Label Transformation (SIMMT) applied to the training data. This transformation converted single-label data into multi-label by assessing vocabulary similarities among dialects. They then employed an ensemble approach that combined predictions from multiple fine-tuned models, namely ArBERT, MARBERT, and MARBERTv2, to enhance overall performance. Their approach achieved an F1-score of 50.57%. The superior performance of this approach can be attributed to two key innovations, the SIMMT transformation, which effectively addressed the challenge of multi-label classification by enriching the training data with dialectal similarities, and the ensemble of multiple BERT-based models, which leveraged the strengths of diverse pre-trained language models to improve robustness and generalization. In contrast, Kanjirangat et al.'s [102] unsupervised cross-encoder with post-filtering, while effective, relied solely on MARBERTv2 and did not incorporate additional data enrichment or ensemble techniques. This highlights the importance of data transformation and model ensembling in achieving state-of-the-art performance.

In a related vein, a three-stage neural models were proposed in another study by Abdelmajeed et al. [124], employing LSTM- and Transformer-based architectures, and integrating attention mechanism to improve ADI in various benchmarks. They achieved the best performance using BERT-based models across benchmark datasets. For MADAR-2, ArBERT-based models performed highest with 98.37% accuracy and an F1 score of 87.98%. In MADAR-6 and MADAR-26, CAMeLBERT excelled with top scores of 91.20% (F1: 91.21%) and 61.66% (F1: 61.59%). In the unbalanced MADAR-9, CAMeLBERT showed the highest accuracy at 79.60%, but ArBERT-based models achieved a higher F1 score (75.75%). For NADI 2020, CAMeLBERT reached the highest accuracy with a significantly better F1 score of 25.94%, outperforming MDA-BERT's 24.81%. Finally, MDA-BERT performed best on QADI, with an accuracy of 67.16% (F1: 66.85%).

In the MADAR shared task, Elaraby et al. [74] proposed a character-level Convolutional BiLSTM model. Their approach involved a convolutional layer to extract local character-level features, followed by a BiLSTM to capture long-range dependencies within text sequences. They trained their model on approximately 2k users' data, where each user's tweets were concatenated into a single input sequence. This approach allowed the model to capture contextual information across a user's tweets, enhancing dialect classification performance. Their system achieved an F1-score of 61.54% and an accuracy of 72.6%, which, while competitive, falls slightly short of Abdelmajeed et al.'s [124] CAMeLBERT-based model in terms of F1-score (61.59% vs. 61.54%).

Another approach was introduced by Alsuwaylimi [33], explored combining BiLSTM layers with CAMeLBERT and ALBERT models independently, each model paired with its own BiLSTM

layers and trained on a dataset of user-generated comments. In this setup, the CAMeLBERT-BiLSTM model achieved the highest performance, with an accuracy of 87.67% and an F1=-score of 87.76%, while the ALBERT-BiLSTM model achieved an accuracy of 86.51% and an F1-score of 86.69%.

Hassan et al. [80] explored BERT-based models for multi-dialect ADI. The models were applied to three tasks: dialect identification, sentiment analysis, and topic classification. The datasets consist of over 63k tweets across six dialects, comparing the performance of DarijaBERT, MARBERT, and DziriBERT against traditional methods. Their results revealed DziriBERT as the top performer, achieving a 87% accuracy in ADI.

Table 3.1 summarizes the key characteristics of the related works discussed above.

Work	**Data**		**Model**	**Class**	**Accuracy %**	**F1-score %**
Traditional Machine Learning Approaches						
Aliwy et al. [23]	NADI 2020		Voting mechanism (Naive Bayes, Logistic Regression, and Decision Trees)	21	—	**20.05**
Ali et al. [126]	NADI 2021	Subtask 1.1	CNB	21	**23.24**	**12.99**
		Subtask 1.2		21	**37.16**	**18.72**
		Subtask 2.1		100	**3.38**	**3.51**
		Subtask 2.2		100	**4.8**	**4.55**
Jauhiainen et al. [96]	NADI 2022	ADI	Naive Bayes	18 (Train & Test-A) k (Test-B)	—	**19.63** on Test-A **10.58** on Test-B
Mishra et al. [123]	MADAR	Subtask 1	Multinomial Naive Bayes	26	**66.31**	**66.21**
		Subtask 2	SVM	21	**67.20**	**57.90**
Talafha et al. [148]		Subtask 2	LinearSVC	21	**76.20**	**69.86**
Baimukan et al. [42]		Subtask 1	MNB with Aggregated Classifiers	25	**77.12**	**74.83**

Work	Data		Model	Class	Accuracy %	F1-score %
El-Haj et al. [70]	Arabic Dialects Dataset[1]		SVM	5	**76** on test data **66** on unseen data	—
Deep Learning Approaches (Non-Transformer)						
Issa et al. [93]	NADI 2021	Subtask 1.2	LSTM	21	**41.36** on Dev set	**22.10** on Dev set
AlShenaifi et al. [31]	NADI 2022	ADI	BiLSTM	18 (Train & Test-A) k (Test-B)	**39.9** on Test-A **23.7** on Test-B	**22.4** on Test-A **9.3** on Test-B
De Francony et al. [66]	MADAR	Subtask 1	Hierarchical BiLSTM + Voting Ensemble	26		**63.02**
Fares et al. [78]		Subtask 1	Ensemble CharTFIDF + WordTFIDF + NN & MNB	26	**65.75**	**65.35**
Transformer-Based Approaches (Transfer Learning)						
El Mekki et al. [72]	NADI 2021	Subtask 1.1	MARBERT-based Multi-Task Learning model	21	**33.84**	**21.4**
		Subtask 1.2		21	**50.30**	**30.64**
		Subtask 2.1		100	**5.72**	**5.35**
		Subtask 2.2		100	**7.92**	**7.32**
AlKhamissi et al. [24]		Subtask 1.1	Ensemble of Different MARBERT Configurations	21	**35.72**	**22.38**
		Subtask 1.2		21	**51.66**	**32.26**
		Subtask 2.1		100	**6.66**	**6.43**
		Subtask 2.2		100	**9.46**	**8.60**
Humayun et al. [90]		Subtask 1.2	Ensemble MARBERT	21	**53.96** on Dev set	—
AlHassan et al. [40]	NADI 2022	ADI	Multi-task AraBERT Model	18 (Train & Test-A) k (Test-B)	—	**32.63** on Test-A **15.61** on Test-B

[1]Available on http://www.lancaster.ac.uk/staff/elhaj/corpora.htm

Work	Data		Model	Class	Accuracy %	F1-score %
Bayrak et al. [47]	NADI 2023	ADI	MARBERTv2	18 (Train & Test-A) k (Test-B)	**51.66** on Test-A **33.89** on Test-B	**34.87** on Test-A **17.19** on Test-B
			MARBERT		**49.18** on Test-A **35.14** on Test-B	**31.66** on Test-A **17.51** on Test-B
Shammary et al. [138]		ADI	Fine-tuned MARBERT	18 (Train & Test-A) k (Test-B)	**32.18** on Test-A **33.38** on Test-B	**18.62** on Test-A **16.68** on Test-B
			Adapter-based MARBERT with data augmentation		**11.52** on Test-A **33.92** on Test-B	**4.85** on Test-A **17.67** on Test-B
Nwesri et al. [127]		ADI	MARBERTv2	18	**82.86**	**82.87**
Adel et al. [14]		ADI	MARBERT Ensemble CNN	18	**83.67**	**83.73**
Elkaref et al. [75]		ADI	NLPeople	18	**87.22**	**87.27**
Kanjirangat et al. [102]	NADI 2024	MLDID	Un-Cross+ Post-FiltN	18 (Train) 8 (Dev) k (Test)	**71.88**	**43.27**
Karoui et al. [103]		MLDID	SIMMT + Ensemble (ArBERT, MARBERT, & MARBERTv2)	18 (Train) 8 (Dev) k (Test)	—	**50.57**
Abdelmajeed et al. [124]	MADAR-2		ArBERT-based	2	**98.37**	**87.98**
			CAMeL-based		**98.21**	**86.67**
			MDA-BERT-based		**98.25**	**86.67**
	MADAR-6		ArBERT-based	6	**91.09**	**91.10**
			CAMeL-based		**91.20**	**91.21**
			MDA-BERT-based		**91.20**	**91.21**
	MADAR-9		ArBERT-based	9	**79.28**	**75.75**
			CAMeL-based		**79.60**	**75.42**

Work	Data		Model	Class	Accuracy %	F1-score %
			MDA-BERT-based		**78.60**	**75.42**
	MADAR-26		ArBERT-based	26	**57.57**	**57.45**
			CAMeL-based		**61.66**	**61.59**
			MDA-BERT-based		**58.71**	**58.71**
	NADI 2020		ArBERT-based	26	**41.26**	**23.29**
			CAMeL-based		**42.31**	**25.94**
			MDA-BERT-based		**42.32**	**24.81**
	QADI		ArBERT-based	26	**60.73**	**60.47**
			CAMeL-based		**65.81**	**65.70**
			MDA-BERT-based		**67.16**	**66.85**
Elaraby et al. [74]	MADAR	Subtask 1	Character-level Convolutional BiLSTM	26	**72.6**	**61.54**
Alsuwaylimi [33]	Arabic Dialect Dataset		CAMeLBERT with BiLSTM	4	**87.67**	**87.76**
			ALBERT with BiLSTM		**86.51**	**86.69**
Hassan et al. [80]	MSDA &Newspaper articles		DarijaBERT	6	**83**	**82**
			MARBERT		**81**	**84**
			DziriBERT		**87**	**85**

Table 3.1: Summary of Dialect identification studies.

3.2 Related Work in Sentiment Analysis

Sentiment analysis is a crucial task in NLP, aimed at discerning the emotional tone behind textual content. In the case of Arabic, the task becomes particularly challenging due to several unique factors. The rich morphological structure of the language, the diversity of dialects, and the frequent occurrence of code-switching all contribute to increased complexity. Moreover, the

limited availability of large-scale, annotated datasets further complicates the development of robust sentiment analysis systems.

Over time, research in Arabic sentiment analysis has evolved across three methodological paradigms. Early studies relied on traditional machine learning techniques, using handcrafted features and lexicon-based approaches to capture sentiment cues. With the advent of DL, models like convolutional and recurrent neural networks enabled automatic extraction of complex patterns from text. More recently, transformer-based approaches have emerged as a dominant paradigm, leveraging pre-trained language models to better capture contextual nuances and improve classification performance.

The following sections discuss key contributions within these paradigms, highlighting their methodologies, strengths, and limitations while tracing the field's progression toward more sophisticated and effective solutions for Arabic sentiment analysis.

3.2.1 Traditional Machine Learning Approaches

In their study on Arabic sentiment analysis, Hadwan et al. [84] investigated user reviews of governmental mobile applications in Saudi Arabia, to enhance service quality. The authors constructed a dataset of 8,000 Arabic user reviews sourced from Google Play, the App Store, and social media platforms. They applied machine learning techniques, including SVM and K-Nearest Neighbors (KNN), to classify sentiments (positive, negative, neutral). Among the evaluated methods, KNN achieved the highest accuracy of 78.46% and an F1-score of 78.96%, outperforming other classifiers. This work highlights the effectiveness of traditional ML methods in low-resource settings, especially where interpretability and computational efficiency are prioritized.

Building on this, Alsemaree et al. [29] conducted a study on Arabic sentiment analysis to understand customer opinions of coffee products using social media data. The authors collected 10,646 Arabic tweets referencing major coffee brands (e.g., Starbucks, Dunkin Coffee), manually annotating them into 3,107 positive, 2,945 negative, and 3,064 neutral reviews. They feature extraction used TF-IDF and Minimum Redundancy Maximum Relevance (MRMR) to address Arabic's morphological complexity. They evaluated four classifiers KNN, SVM, decision tree, and random forest, and proposed a meta-ensemble model combining these via hard and soft voting. The hard-voting ensemble achieved the highest accuracy (95.95%), outperforming individual models (SVM: 95%, RF: 94%, DT: 95%, KNN: 74%). This work leverages ensemble learning to enhance prediction robustness.

Further within the exploration of traditional ML methods, Almuhaya et al. [25] conducted a comparative study on Arabic sentiment Analysis using ML to address imbalanced social media data. The authors used the SS2030 dataset [36], and evaluated the impact of sampling techniques on model performance. They employed five ML algorithms SVM, Logistic Regression, Random Forest, MNB, and KNN alongside preprocessing (text cleaning, normalization) and

TF-IDF for feature extraction. To mitigate class imbalance, Synthetic Minority Over-sampling Technique (SMOTE) and Random Under-Sampling (RU-S) were applied. Results demonstrated that SVM outperformed other models, achieving 85% accuracy without sampling and 84% with SMOTE/RU-S, while KNN performed poorly (65% accuracy).

3.2.2 Deep Learning Approaches (Non-Transformer)

Transitioning to more advanced methodologies, Berrimi et al. [51] proposed a novel attention-based deep learning architecture for Arabic sentiment analysis, combining Bidirectional GRU with an additive attention mechanism. The model leverages FastText embeddings and learnable embeddings to address morphological complexity and dialectal diversity in Arabic. Evaluated on three large-scale datasets LABR [35], HARD [77], and BRAD [76], the model achieved state-of-the-art accuracies of 95.73%, 96.29%, and 95.65%, respectively, and the corresponding F1-scores of 97.13%, 96.28%, 96.15%, outperforming baseline models (BiGRU/BiLSTM). Key innovations include the attention mechanism's ability to focus on sentiment-relevant textual segments and the integration of bidirectional layers for contextual understanding. The authors further demonstrated the model's language- and task-independence by achieving competitive results on Russian and English sentiment analysis, English hate speech detection, and Arabic news categorization, underscoring its generalizability. Statistical validation (t-tests) confirmed significant performance improvements over existing methods.

Continuing the exploration of deep learning, Alwehaibi et al. [34] employed deep learning models to address the challenges posed by MSA and dialectal variations using the AraSenTi dataset, a corpus of 15,945 labeled tweets [16]. They extracted features via FastText word embeddings. Three models were evaluated: an LSTM for sequential word-level context, a CNN for character-level patterns, and an ensemble model combining both (LSTM-CNN). The ensemble model outperformed others, achieving 96.7% accuracy (mean of 10-fold cross-validation).

Further enhancing deep learning approaches, Ahmad et al. [15] proposed a hybrid deep learning model combining CNN and LSTM for sentiment analysis. The study utilized the Arabic Reviews dataset. Preprocessing involved text normalization, stopword removal, and padding to unify sentence lengths. The model architecture included an embedding layer, a CNN layer, an LSTM layer, and dense layers with a sigmoid output for binary classification. The model achieved 95% accuracy, and F1-score, outperforming BiLSTM (91.99%), standalone CNN (92.8%), and SVM-based approaches (84%). The hybrid model's strength likely stemmed from leveraging CNN's local feature extraction and LSTM's sequential context retention, demonstrating the potential of combining multiple neural architectures.

3.2.3 Transformer-Based Approaches (Transfer Learning)

Moving to the state-of-the-art in sentiment analysis, Chouikhi et al. [62] addressed challenges in Arabic sentiment analysis by proposing a BERT-based model with an Arabic tokenizer. Their

approach integrated an Arabic BERT tokenizer and a medium-sized Arabic BERT model (8 encoder layers) with hyperparameter tuning via random search. The model concatenated hidden layers, applied dropout, and used Softmax for classification. Experiments on five datasets ASTD [125], HARD, LABR, AJGT [27], and ArSenTD-Lev [45], showed superior performance, achieving 96.11% accuracy on AJGT and 75% on ArSenTD-Lev, 87% on LABR, outperforming benchmarks like AraBERT and Arabic BERT, on ASTD, and HARD, it achieved 91%, and 95%, respectively, lagging just a bit behind AraBERT. Key improvements stemmed from enhanced tokenization and optimized architecture. However, despite its strong performance, this approach fell short on LABR and HARD compared to the attention-based BiGRU model of Berrimi et al. [51], which achieved 95.73% and 96.29% accuracy on these datasets, respectively. The superior results of the attention mechanism suggest that incorporating explicit weighting of sentiment-relevant textual segments enhances classification, whereas the BERT-based model primarily benefits from refined tokenization and pretraining. This contrast underscores the potential for combining contextual embeddings with attention mechanisms to further optimize sentiment analysis in Arabic.

Building on transformer-based approaches, Mhamed et al. [120] addressed the challenges posed by Arabic's morphological complexity and dialectal diversity by developing two CNN-based models, MC1 and MC2, combined with enhanced text preprocessing. The authors focused on the ASTD and ATDFS datasets [39], evaluating both binary and multiclass classification tasks. MC1 utilized a CNN with global average pooling, followed by a dense layer, while MC2 integrated CNN with max pooling followed by a BiGRU and a dense layer. Additionally, both models leveraged AraBERT to generate word embeddings, enhancing their ability to capture contextual information. Their models achieved the following results : MC1 attained 90.06% accuracy on ASTD (2-class) and MC2 reached 73.17% on ASTD (4-class), outperforming prior benchmarks, as for the ATDFS dataset the MC1 and MC2 achieved 92.63% and 92.96%, respectively. Despite incorporating AraBERT embeddings, their best model on ASTD with 4 classes (MC2) still fell significantly short of Chouikhi et al.'s [62] BERT-based approach, which achieved 91% accuracy. This suggests that fine-tuning a full transformer model, rather than solely using its embeddings, may provide stronger sentiment representation.

Further advancing transformer-based methods, Alosaimi et al. [28] proposed a novel model, ArabBert-LSTM, which combines the transformer-based AraBERT model with LSTM to improve sentiment analysis for Arabic text. The authors addressed the challenges posed by Arabic's complex morphology and dialectal variations by leveraging AraBERT for contextual word embeddings and LSTM for sequence modeling and long-term dependencies. The model was evaluated on four Arabic datasets: SS2030, ASTC [2], Main-AHS [18], and Sub-AHS [18]. The proposed model outperformed traditional ML and DL models, achieving impressive accuracy rates of 90.40% on SS2030, 93.76% on ASTC, 92.61% on Main-AHS, and 97.12% on Sub-AHS. The F1-scores were equally strong, with 90.41%, 93.76%, 92.61%, and 97.10% for the respective datasets. Compared

[2]Available at https://www.kaggle.com/datasets/mksaad/arabic-sentiment-twitter-corpus

to Almuhaya et al. [25], who applied traditional ML models to SS2030, ArabBert-LSTM significantly outperformed the best-performing SVM model (85%) by a notable margin, highlighting the advantage of DL and contextual embeddings over TF-IDF-based ML approaches.

In a similar vein, Bakhit et al. [43] proposed a hybrid neural network model combining RNN and BiLSTM to analyze customer satisfaction in Arabic text. The authors utilized the Arabic Reviews Dataset (ARD) [3] as the source domain and evaluated transfer learning on two target datasets: ASTD and Aracust. They employed AraBERT, a pre-trained Arabic language model, for word embeddings to capture contextual nuances. The RNN-BiLSTM model achieved 95.75% accuracy, and 95.72% F1-score on ARD, outperforming standalone RNN (90.97% and 90.91%) and BiLSTM (92.64% and 92.70%). The model attained 95.44% and 96.19% accuracy on ASTD and Aracust [26], respectively, along with F1-scores of 96.61% and 94.30% on the same datasets, demonstrating robustness across different datasets. Comparatively, Chouikhi et al. [62] employed a BERT-based model with an optimized Arabic tokenizer and hyperparameter tuning, reaching 91% accuracy on ASTD. While their method outperformed several benchmark transformer models, it still lagged behind Bakhit et al.'s RNN-BiLSTM approach, which benefited from sequential modeling and transfer learning. The accuracy gap suggests that BiLSTM's ability to capture long-term dependencies, coupled with AraBERT embeddings, contributes more effectively to sentiment classification than direct fine-tuning of a transformer model with limited encoder depth.

Expanding the scope to Moroccan Arabic, Hannani et al. [86] conducted a study on Moroccan Arabic sentiment analysis, evaluating ChatGPT-4, fine-tuned BERT models, FastText embeddings, and traditional ML classifiers across two datasets, the Moroccan Arabic Twitter Corpus (MAC) [82] and the Moroccan Arabic YouTube Corpus (MYC) [97]. ChatGPT-4 performed poorly, while fine-tuned BERT models, particularly DarijaBERT, achieved strong results on MAC, reaching 90% accuracy and an 87.7% F1-score. When trained and evaluated on MYC, DarijaBERT-Arabizi achieved the highest accuracy and F1-score at 85.6%. FastText embeddings demonstrated strong performance on MAC (83.7% accuracy) but declined to 79% on MYC. Traditional ML classifiers, such as Naïve Bayes and SVM, showed moderate performance on MAC but struggled on MYC.

Further exploring Moroccan Arabic sentiment analysis,the study by Matrane et al. [115] investigates sentiment analysis challenges for the Moroccan Arabic dialect, focusing on the limitations of dialect-specific preprocessing methods and their impact on natural language processing tasks. The research evaluates preprocessing techniques, such as stemming and feature extraction, using two transfer learning approaches — feature extraction with deep learning models and fine-tuning pre-trained models — across four datasets: the Moroccan Sentiment Analysis Corpus (MSAC) [4], Moroccan Sentiment Twitter Dataset (MSTD) [121], a Facebook dataset (FB), and the MAC dataset. The DarijaBERT model emerged as the top performer on

[3]Available at https://www.kaggle.com/datasets/abedkhooli/arabic-100k-reviews
[4]Available at https://github.com/ososs/Arabic-Sentiment-Analysis-corpus

the FB and MSTD datasets, achieving a 93.37% accuracy with an 88.55% F1-score on FB, and a 61.26% accuracy paired with a 53.51% F1-score on MSTD. In contrast, the QARIB model delivered the best outcomes for the MAC and MSAC datasets, recording accuracies of 89.96% and 90.38% and corresponding F1-scores of 88.04% and 90.38%, respectively. On the MAC dataset, Hannani et al. [86] observed that the fine-tuned BERT model, DarijaBERT, reached very similar performance levels, achieving an accuracy of 90% and an F1-score of 87.7%. These comparable results suggest that both approaches are highly effective for sentiment analysis in Moroccan Arabic, with only minor variations likely attributable to differences in preprocessing and model fine-tuning strategies.

Shifting the focus towards generative models, Salma Khaled et al. [105], investigated the performance of generative Large Language Models (LLMs) enhanced by Retrieval-Augmented Generation (RAG) architecture for Arabic Sentiment Analysis across three benchmark datasets: ASAD [21], ArSarcasm-v2 [12], and SemEval [134]. The authors' RAG model initially scored 59%, 57%, and 64% F1-scores respectively. Challenges such as dataset imbalances and misclassified neutral labels were identified, prompting the removal of the neutral class, which significantly boosted the RAG model's performance to 76% (ASAD), 75% (ArSarcasm-v2), and 82% (SemEval) F1-scores.

In their study on Arabic sarcasm detection and sentiment analysis, using ArSarcasm-v2 dataset, Alharbi et al. [20] proposed a multi-task learning approach combining Static Word Embeddings (SWE) and contextualized embeddings (MARBERT) to improve performance on both tasks. The system utilized Ara2Vec and Character-level Embeddings (CE) for SWE, alongside MARBERT, to capture semantic and contextual nuances. The proposed MTL-CNN-LSTM architecture concatenated these embeddings and achieved competitive performance in sentiment analysis, with an F-PN of 70.1%.

Hengle et al. [88] proposed a hybrid model for Arabic sarcasm detection and sentiment identification, combining contextualized representations from AraBERT with SWE trained on Arabic social media corpora. The proposed system achieved an F-PN score of 70.73%, outperforming baseline models, including standalone AraBERT. Both of this study and Alharbi et al. [20] leverage a combination of static and contextual embeddings, yet they differ in their embedding choices and architectures. Alharbi et al. [20] employ a multi-task learning CNN-LSTM framework that concatenates Ara2Vec and CE with contextualized embeddings from MARBERT, achieving an F-PN of 70.1%. In contrast, Hengle et al. integrate contextualized representations from AraBERT with SWE derived from Arabic social media corpora, resulting in a slightly higher F-PN score of 70.73%. This slight performance gap suggests that the specific choice of contextual model (AraBERT versus MARBERT) may enhance the model's ability to capture the subtle linguistic features associated with sentiment.

Expanding on ensemble techniques, Song et al. [143] presented a deep ensemble-based method for sentiment detection, using ArSarcasm-v2 dataset. The system leverages fine-tuned

pre-trained language models, including XLM-R and AraBERT, combined with task-adaptive pre-training (TAPT) and knowledge distillation to enhance performance. A stacking mechanism, using SVM, was employed to aggregate predictions from multiple models. The system achieved competitive results, an F-PN of 73.92%. Compared to Hengle et al. [88], whose hybrid approach fused AraBERT-based contextualized embeddings with SWE to achieve an F-PN of 70.73%, Song et al. achieved a higher F-PN of 73.92%. This performance boost underscores the advantage of leveraging diverse model architectures and advanced ensemble techniques over a more straightforward hybrid embedding approach.

In a complementary approach, El Mahdaouy et al. [71] proposed a deep multi-task learning model for sarcasm detection and sentiment analysis in Arabic, leveraging the MARBERT language model and a multi-task attention interaction module. The model integrates task-specific attention layers and a Sigmoid interaction layer to enable knowledge sharing between sarcasm detection and sentiment analysis tasks. The proposed system achieved competitive results, with an F-PN score of 74.80% on sentiment analysis. Both of this approache and Song et al.'s [143] one aim to enhance sentiment detection on the ArSarcasm-v2 dataset and yielded competitive results, this approach achieved a slightly higher F-PN score (74.80%) compared to Song et al. (73.92%). This marginal improvement suggests that the incorporation of multi-task learning with task-specific attention may better capture sentiment nuances than an ensemble approach relying on stacking multiple models. The difference, while small, points to the potential benefits of shared representations and task-specific attention mechanisms in enhancing performance on multi-task models.

Lastly, Kaseb et al. [104] introduced active learning, focusing on sentiment analysis using the ArSarcasm-v2 dataset. Active learning, which involves selectively labeling data to maximize model performance, was applied to address the scarcity of labeled Arabic datasets. The study utilized the SAIDS (Sentiment Analysis Informed of Dialect and Sarcasm) model, that predicts sarcasm and dialect and then uses them to predict the sentiment of the text, multiple active learning experiments were conducted with varying query strategies, the results demonstrated that active learning significantly reduces the amount of labeled data required for training while maintaining high performance. Specifically, the study achieved a state-of-the-art F1-score of 76.71% for sentiment analysis using 95% of the dataset, highlighting that some data points can degrade performance. The paper concludes that active learning is a powerful tool, especially given the limited availability of labeled Arabic datasets.

Table 3.2 summarizes the key characteristics of the related works discussed above.

Work	**Data**	**Model**	**Class**	**Accuracy %**	**F1-score %**
		Traditional Machine Learning Approaches			

Work	Data	Model	Class	Accuracy %	F1-score %
Hadwan et al. [84]	Collected 8k Arabic user reviews	KNN	3	**78.46**	**78.96**
Alsemaree et al. [29]	Collected 10,646 Arabic tweets	Ensemble (Hard votig)	3	**95.95**	—
Almuhaya et al. [25]	SS2030	SVM	2	**85**	**85**
Deep Learning Approaches (Non-Transformer)					
Berrimi et al. [51]	LABR	BiGRU with Additive attention	2	**95.73**	**97.13**
	HARD		2	**96.29**	**96.28**
	BRAD		2	**95.65**	**96.15**
Alwehaibi et al. [34]	AraSenTi	Ensemble LSTM-CNN	3	**96.7**	—
Ahmad et al. [15]	Arabic Reviews dataset	Hybrid LSTM-CNN	2	**95**	**95**
Transformer-Based Approaches (Transfer Learning)					
Chouikhi et al. [62]	ASTD	BERT-based model with Arabic tokenizer	4	**91**	—
	HARD		3	**95**	—
	LABR		2	**87**	—
	AJGT		2	**96.11**	—
	ArSenTD-Lev		3	**75**	—
Mhamed et al. [120]	ASTD	MC1	2	**90.06**	—
		MC2		**89.49**	—
		MC1	4	**72.43**	—
		MC2		**73.17**	—
	ATDFS	MC1	2	**92.63**	—
		MC2		**92.96**	—

Work	Data	Model	Class	Accuracy %	F1-score %
Alosaimi et al. [28]	SS2030	ArabBert-LSTM	2	**90.40**	**90.41**
	ASTC		2	**93.76**	**93.76**
	Main-AHS		2	**92.61**	**92.61**
	Sub-AHS		2	**97.12**	**97.10**
Bakhit et al. [43]	ARD	AraBERT-RNN-BiLSTM		**95.75**	**95.72**
	ASTD		4	**95.44**	**96.61**
	Aracust		2	**96.19**	**94.30**
Hannani et al. [86]	MAC	DarijaBERT	3	**90**	**87.7**
	MYC	DarijaBERT-Arabizi	2	**85.6**	**85.6**
Matrane et al. [115]	MSAC	QARIB	2	**90.38**	**90.38**
	MSTD	DarijaBERT	4	**61.26**	**53.51**
	FB	DarijaBERT	2	**93.37**	**88.55**
	MAC	QARIB	3	**89.96**	**88.04**
Salma Khaled et al. [105]	ASAD	RAG Model	2	—	**76**
			3	—	**59**
	ArSarcasm-v2		2	—	**75**
			3	—	**57**
	SemEval		2	—	**82**
			3	—	**64**
Alharbi et al. [20]	ArSarcasm-v2	MTL-CNN-LSTM	3	—	**F-PN 70.1**
Hengle et al. [88]	ArSarcasm-v2	AraBERT + CNN-BiLSTM	3	**68.40**	**F-PN 70.73** **F1-macro 62.32**
Song et al. [143]	ArSarcasm-v2	XLM-R + AraBERT + SVM	3	—	**F-PN 73.92**

Work	Data	Model	Class	Accuracy %	F1-score %
El Mahdaouy et al. [71]	ArSarcasm-v2	MTL ATTINTER	3	—	**F-PN 74.80**
Kaseb et al. [104]	ArSarcasm-v2	SAIDS	3	—	**76.71**

Table 3.2: Summary of Sentiment Analysis studies.

3.3 Comparative Analysis

The diverse work in Arabic Dialects highlights a clear evolution in methodology, from traditional ML relying on engineered features techniques to DL approaches that learn representations automatically, and finally to transformer-based models that leverage large-scale pre-training and transfer learning. Below, we compare these paradigms in terms of their strengths, limitations, and performance characteristics as observed in recent shared tasks and individual studies.

3.3.1 Traditional ML vs. Deep Learning (Non-Transformer)

Traditional ML techniques have been widely applied to both ADI and sentiment analysis. These methods typically rely on manually engineered features, such as n-grams, TF-IDF, and lexical resources, combined with classifiers like Naïve Bayes, SVM, and Random Forest. For example, in ADI, studies like those by Aliwy et al. [23] and Ali et al. [126] demonstrated the effectiveness of statistical classifiers, achieving moderate F1-scores (e.g., 20.05% and 18.72%, respectively) on shared task datasets. Similarly, in sentiment analysis, Hadwan et al. [84] and Alsemaree et al. [29] showed that traditional ML models, such as KNN and ensemble methods, could achieve competitive accuracies (e.g., 78.46% and 95.95%, respectively).

The primary strengths of traditional ML approaches lie in their computational efficiency, interpretability, and ability to perform well with limited data. For instance, Mishra et al. [123] demonstrated that an MNB classifier, combined with word and character-level n-grams, achieved an accuracy of 66.31% and an F1-score of 66.21% on the MADAR-26 dataset. This highlights the effectiveness of traditional ML in low-resource settings, where annotated data is scarce. Similarly, Talafha et al. [148] showed that a LinearSVC classifier, when combined with a dialect weighting scheme, achieved an accuracy of 76.20% and an F1-score of 69.86%, further underscoring the potential of traditional ML methods when paired with well-designed feature engineering techniques.

However, these methods are inherently constrained by the quality of handcrafted features, which often fail to capture the deep morphological, syntactic, and contextual nuances of Arabic dialects and sentiment expressions. For example, while traditional ML models perform well on

high-level tasks, they struggle with fine-grained dialect identification and sentiment analysis, particularly in the presence of dialectal variations, code-switching, and nuanced expressions. This limitation is evident in the NADI shared tasks, where traditional ML models often face challenges in achieving high performance, especially in fine-grained classification tasks. For instance, in the NADI 2021 shared task, Ali et al. [126] reported F1-scores of 12.99% and 18.72% for country-level classification, but performance dropped significantly for province-level classification, with F1-scores of 3.51% and 4.55%. This suggests that traditional ML methods, while effective for broader tasks, struggle with the complexity of fine-grained dialect identification.

This shortcoming illustrates a key bottleneck of traditional ML: its dependence on shallow lexical features rather than deep semantic or syntactic structures. Consequently, these approaches tend to generalize poorly when faced with unseen dialectal variations or informal language, limiting their adaptability and robustness in real-world applications.

In contrast, deep learning (non-transformer) approaches, such as CNNs, LSTMs, and BiLSTMs, automate the feature extraction process, enabling models to learn hierarchical representations directly from raw text. For example, in ADI, Issa et al. [93] and AlShenaifi et al. [31] demonstrated that LSTM and BiLSTM models outperform traditional ML methods, achieving F1-scores of 22.10% and 22.4%, respectively. Similarly, in sentiment analysis, Berrimi et al. [51] and Alwehaibi et al. [34] showed that attention-based BiGRU and hybrid LSTM-CNN models achieve state-of-the-art accuracies of 96.29% and 96.7%, respectively.

The key advantage of deep learning lies in its ability to capture complex patterns and long-range dependencies, which are crucial for tasks like ADI and sentiment analysis. However, these models require larger datasets and more computational resources compared to traditional ML methods. Additionally, while they offer superior performance, their "black-box" nature reduces interpretability, making it harder to understand the decision-making process.

3.3.2 Traditional DL vs. Transformer (Transfer Learning)

While traditional deep learning models excel at capturing sequential and local patterns, they often struggle with global contextual understanding, especially in tasks involving long-range dependencies or fine-grained distinctions. Transformer-based models address these limitations by leveraging self-attention mechanisms, which enable them to model relationships between all tokens in a sequence effectively. This capability is particularly beneficial for Arabic, given its rich morphology, dialectal diversity, and contextual nuances.

In ADI, transformer-based models like MARBERT and AraBERT have set new benchmarks. For example, El Mekki et al. [72] and AlKhamissi et al. [24] demonstrated that fine-tuning MARBERT with task-specific adaptations (e.g., multi-task learning and vertical attention) significantly improves performance, achieving F1-scores of 30.64% and 32.26% on country-level classification tasks. Similarly, in sentiment analysis, transformer-based models like AraBERT and ArabBert-LSTM have demonstrated strong performance on benchmark datasets. For exam-

ple, Chouikhi et al. [62], and Alosaimi et al. [28] reported that their BERT-based model achieved state-of-the-art results highlighting the ability of transformer-based models to capture contextual nuances and long-range dependencies, which are critical for NLP in Arabic.

However, it is important to note that traditional deep learning models remain highly competitive in many scenarios. For instance, Berrimi et al. [51] demonstrated that an attention-based BiGRU model achieved state-of-the-art accuracies of 95.73% and 96.29% on the LABR and HARD datasets, respectively, outperforming some transformer-based models such as Chouikhi et al. [62]. This suggest that traditional deep learning models, particularly when enhanced with attention mechanisms or hybrid architectures, can still achieve competitive results, especially in tasks where local patterns and sequential dependencies are more important than global context.

The primary strength of transformer-based models lies in their ability to leverage large-scale pre-training and transfer learning. By pre-training on massive corpora, these models develop robust contextual embeddings that can be fine-tuned for specific tasks with relatively small datasets. This approach not only enhances performance but also reduces the dependency on large annotated datasets, which are often scarce for Arabic dialects. However, transformer-based models come with significant computational costs and require careful hyperparameter tuning. Their complexity also makes them less interpretable compared to traditional ML and non-transformer deep learning models. Despite these challenges, their superior performance across various benchmarks underscores their effectiveness in handling the linguistic intricacies of Arabic.

3.4 From Limitations to Direction: Gaps in Current Approaches and Motivations for This Research

The comprehensive review of existing works in tasks such as ADI and sentiment analysis highlights the broader challenges faced by low-resource languages, particularly Arabic dialects. These tasks serve as illustrative examples of the difficulties posed by data scarcity, dialectal diversity, and the need for innovative solutions to address the unique linguistic characteristics of under-resourced dialects. A key opportunity lies in leveraging the wealth of resources available in high-resource settings, such as MSA, to bridge the gap for low-resource dialects like Algerian Arabic. By utilizing pre-trained models, transfer learning techniques, and leveraging the wealth of data available in MSA, this thesis aims to mitigate data scarcity and improve performance in under-resourced settings. This section synthesizes the insights gained from the literature review and outlines how they have guided the research trajectory of this thesis.

3.4.1 Identifying Gaps and Challenges

The literature highlights several persistent challenges that remain unresolved in low-resource language processing, particularly for Arabic dialects:

- **Data Scarcity**: Annotated datasets for Arabic dialects are limited, especially for fine-grained tasks such as province-level ADI or nuanced sentiment classification. This scarcity hinders the development of robust models, particularly for low-resource dialects like Algerian Arabic.

- **Dialectal Complexity**: The rich morphological and lexical diversity of Arabic dialects, coupled with frequent code-switching, poses significant challenges for modeling and generalization. Traditional and deep learning models often struggle to capture these complexities effectively.

The progression from traditional ML to DL and transformer-based approaches reflects a shift toward models that are increasingly capable of handling the complexities of Arabic dialects and sentiment analysis. Traditional ML methods provide a solid baseline with lower computational demands and better interpretability, making them suitable for low-resource scenarios. Deep learning models offer enhanced performance through automated feature extraction and sequential modeling, though at the cost of increased computational requirements and reduced interpretability. Transformer-based models represent the state-of-the-art, leveraging global context and transfer learning to achieve superior performance, though they come with higher computational complexity and resource demands.

This evolution highlights the trade-offs between simplicity, performance, and resource requirements. Our research focuses on bridging these gaps, by developing more efficient transformer architectures, transferring knowledge from high-resource to low-resource settings to address data scarcity and improve performance in under-resourced dialects and tasks.

3.5 Conclusion

The literature review has provided a solid foundation for identifying the key challenges and opportunities in low-resource language processing, with a particular focus on Arabic dialects. By addressing these challenges through innovative methodologies and leveraging the strengths of existing approaches, this thesis aims to bridge the gap between high-resource and low-resource settings. The proposed contributions not only address the limitations of current models but also pave the way for future research in this dynamic and evolving field. This work has broader implications for the field of NLP, particularly in the context of multilingual and low-resource settings. The methodologies developed here can be adapted to other languages and tasks, providing a framework for leveraging high-resource data and models to support under-resourced languages worldwide. By addressing the challenges of data scarcity, dialectal diversity, this thesis contributes to the ongoing effort to make NLP technologies more inclusive and accessible, ensuring that the benefits of AI and machine learning are available to all languages and communities. To advance this vision, we put forward the assumption that linguistic knowledge acquired from MSA, supported by abundant annotated corpora and robust pretrained models, can be leveraged

to improve the understanding and modeling of dialectal Arabic, particularly in low-resource settings such as Algerian Arabic. This assumption guides the design and evaluation of the approaches developed throughout the thesis, beginning with the next chapter, which presents a set of preliminary experiments exploring traditional machine learning and transfer learning techniques for dialectal NLP. These early explorations serve as feasibility studies and establish critical baselines that inform the development of more advanced hybrid models in the subsequent chapters.

CHAPTER 4

EXPLORATORY ANALYSIS OF TRADITIONAL MACHINE LEARNING AND TRANSFER LEARNING FOR DIALECTAL NLP

This chapter investigates how well models can handle dialectal Arabic under low-resource conditions, starting from a scenario with no prior linguistic knowledge. The goal is to evaluate whether traditional machine learning models, relying solely on surface-level features, are capable of capturing meaningful patterns in dialectal text. If models with no exposure to linguistic structure can already learn something useful, this raises the question: can we achieve even better results by leveraging knowledge from a closely related, high-resource variety like MSA? To answer this, the chapter is structured in two-step exploration, it uses two case studies: a disaster-related tweet classification task and a dialect classification task using MSA-based transfer learning. The development of NLP models for Arabic dialects presents specific challenges, including data scarcity, dialectal diversity, and the absence of standardized orthography. While deep learning models have achieved strong results in high-resource languages, they often require substantial annotated data, which is typically unavailable for dialects like Algerian Arabic. Consequently, alternative strategies must be considered.

The first case study investigates the effectiveness of traditional machine learning models for dialectal classification without relying on any pretrained linguistic features. These models, trained on handcrafted statistical features, are evaluated on a dataset of disaster-related tweets in dialectal Arabic. The goal is to determine whether meaningful linguistic patterns can be captured without any prior knowledge, and whether such simple models are sufficient for tasks involving noisy, real-world dialectal data.

The second part of the chapter builds on these findings by examining whether pretrained models trained on MSA can generalize to dialectal inputs. Specifically, we test the performance of MSA-based transformer models on a dialect classification task, allowing us to assess the extent

to which linguistic knowledge from a closely related variety can be transferred to improve model performance in a low-resource setting.

Together, these experiments establish essential performance baselines and provide insight into the capabilities and limitations of both knowledge-free and transfer-based approaches. These insights lay the groundwork for the next phase of our research: leveraging MSA as a linguistic bridge to improve low-resource NLP applications.

Publication Note

The work presented in this chapter has produced two peer-reviewed international conference publications:

- **Part I (4.1)**: M. Chabane, F. Harrag, and K. Shaalan, "Beyond Deep Learning: A Two-Stage Approach to Classifying Disaster Events and Needs," *2024 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pp. 1–7, 2024, doi: 10.1109/ICT-DM62768.2024.10798928.
- **Part II (4.2)**: M. Chabane, F. Harrag, K. Shaalan, and S. Hamdi, "Bridging the Gap: Transfer Learning for Dialect Identification in Low-Resource Settings A Case Study with Algerian Arabic," *2025 International Symposium on iNnovative Informatics of Biskra (ISNIB)*, Biskra, Algeria, 2025, pp. 1–6, doi: 10.1109/ISNIB64820.2025.10982839.

4.1 Disaster Classification as a Case Study

In recent years, the world has been a witness to numerous natural and man-made disasters, ranging from floods and pandemics to bombings and explosions. These crises pose a significant threat to human life, infrastructure, and often disrupt standard communication at the very moment where there is an elevated demand and need for information yet limited supply, within these challenges social media platforms emerge as a lifelines offering real-time updates regarding shelter, help and food, safety measures, as well as reporting damage to life and infrastructure, communicating information about the injured and seek help, a way for the striving people to reach out to loved ones in and out of the disaster area, and empowers those in need to send out rescue calls.

An illustration of the life-saving potential of social media unfolded during the deadly earthquake that struck Turkey and Syria, on February 6, 2023, In the aftermath of the disaster, victims trapped under debris turned to social media in a desperate plea for help, sending out tweets with their their precise location, sharing them with users with large followers [142], and despite the potentially life-saving information embedded within these tweets, a significant challenge looms large: The vast amount of unstructured social media, data generated during disasters on one

hand and the lack of efficient systems capable of processing this data on the other, leads to vital information going unnoticed amidst the digital noise.

Our research endeavors to develop robust methodologies for extracting valuable information from disaster-related tweets. In particular, we focus on two key tasks: identifying the nature of the disaster and extracting humanitarian needs.

Identifying the nature of the disaster is needed especially in scenarios where multiple events occur concurrently, such as the Beirut explosion coinciding with the COVID-19 pandemic. By accurately classifying the type of disaster discussed in each tweet, we aims to facilitate targeted response efforts tailored to the specific challenges posed by each crisis. Additionally, understanding the nature of the disaster helps us see potential overlaps or intersections between different crises, giving us a clearer picture of what's going on and helping us make smarter decisions. Moreover, extracting humanitarian needs helps directing resources and aid to where they are most urgently required. By analyzing the content of disaster-related tweets, we aim to identify and categorize various humanitarian concerns, including requests for assistance and reports of infrastructure damage. This information can be used by emergency responders and aid organizations, guiding their actions and interventions in a manner that maximizes impact and effectiveness.

In the face of disaster, the clock starts ticking immediately. The needs of affected communities—from locating missing people, getting help for the injured ones to securing basic necessities—hold the highest priority and require fast actions. To address this urgency, we explore traditional machine learning models known for their speed and efficiency. These models offer advantages such as quick processing and straightforward implementation, making them suitable for rapid response scenarios. However, as we delve deeper into the complexities of disaster-related tweets, we also investigate the effectiveness of more advanced techniques, such as BERT [67], a state-of-the-art natural language processing model. Specifically, through a comparative analysis of traditional models against the advanced capabilities of BERT. We aim to identify the optimal approach, one that strikes a critical balance between speed and accuracy, ensuring that our methodologies are not only fast in their response but also deeply effective in addressing the urgent needs of disaster response and humanitarian efforts.

In summary, our work aims to unlock the lifesaving potential of social media during disasters. By developing innovative methods for extracting actionable insights from the digital noise of social media, allowing for faster responses to urgent needs in the midst of crisis, ultimately saving lives and alleviating suffering.

4.1.1 Related works

Social Media generally and Twitter specifically has become a go-to tool for emergency communication in recent years, and Several studies analyzes the microblogs collected during various crisis events.

For instance Basu et al. [46], used tweets collected during two major earthquake events. The data was used to classify tweets into three classes: need-tweets, availability-tweets, and other-tweets, with supervised classification approaches and unsupervised retrieval methodologies being employed for this task. The research proposed neural retrieval models that combine word-level embeddings and character-level embeddings for identifying resource needs and availabilities during disaster events. The main results indicated that classification approaches performed better with the availability of good quality training data, while unsupervised retrieval methods outperformed in scenarios where such data were lacking.

In the study conducted by Alam et al. [17], using publicly available datasets such as CrisisLex[1] and CrisisNLP[2], they developed an automatic data processing service which takes in text message e (e.g., a tweet or a Facebook post or an SMS) from various sources and classify it into disaster type, informativeness of the text to the humanitarian aid and for and humanitarian information type. The proposed methodology involved developing classification models using both traditional and deep learning algorithms. The main achieved results indicated that the classification models outperformed existing publicly available models.

The paper written by Kundu et al. [110] focuses on classifying tweets from disaster scenarios into predefined action classes using datasets specific to tweets posted during the Nepal earthquake, that were obtained from Forum for Information Retrieval Evaluation 2016 (FIRE2016) and 2017 (FIRE2017). The proposed methodology involves using deep learning techniques, to classify the tweet data. Results shows that the Deep LSTM model outperformed other methods in terms of Precision, Recall, and F-Score, especially in the FIRE2017 dataset, which contained only 2 classes.

Alharbi et al. [19] investigates crisis-related tweets, and focus on two tasks the first is to classify these tweets as relevant or irrelevant, the second one is to classify those found as relevant onto one or more information categories such as infrastructure damage, caution, etc. Their approach is a selection-based domain adaptation technique, which selects the most similar data points from past crises to train a model for classifying tweets from a new, emerging crisis. This method outperforms training on all past data.

4.1.2 Data description and preprocessing

The IDRISI-D [145] dataset consists of tweets discussing 26 distinct disaster events, including floods, earthquakes, fires, and other crises, occurring across regions where both English and Arabic are spoken. Of particular interest to our study are the Arabic tweets, which capture a different crises across the Arabic-speaking world. These crises are the following COVID-19, floods in Kuwait, Jordan and Hafr Al-Batin Dragon storms, the Beirut explosion, and the Cairo bombing.

[1] https://crisislex.org/
[2] https://crisisnlp.qcri.org/

The dataset has a rich array of information, including timestamps and location mentions, and various features. However, our study focuses primarily on two aspects discussed in the tweet.

- **Nature of the disaster** The tweets are classified into one of the following disaster types: flood, explosion, bombing, Covid-19, or storm.

- **Humanitarian categories** Outlined by various forms of information such as casualties, injuries, displaced individuals, missing or found people, infrastructure and utility damage that can help both government agencies and humanitarian organizations in prioritizing their help and rescue operations.

4.1.2.1 Dataset challenges

- **Limited Data Availability & Imbalanced data** The dataset comprises only 2,170 examples, and is imbalanced, this occurs when the distribution of classes or labels within the dataset is skewed i.e. it has an imbalanced distribution of the examples of different classes.
  This issue appears in both of the two aspects we are focusing on, in Figure 4.1 is a visual representation illustrating the distribution of disaster classes within the dataset.

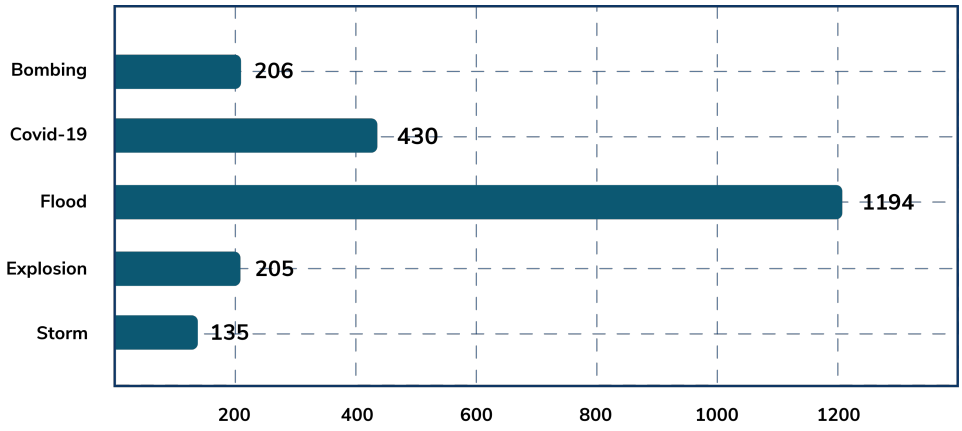

Figure 4.1: Distribution of Disaster Type Classes

In Figure 4.2, we present the frequency of transitions between humanitarian categories within the dataset. Specifically, the plot depicts how often one class of humanitarian categories transitions to another. The y-axis denotes the current class, while the x-axis represents the next class.

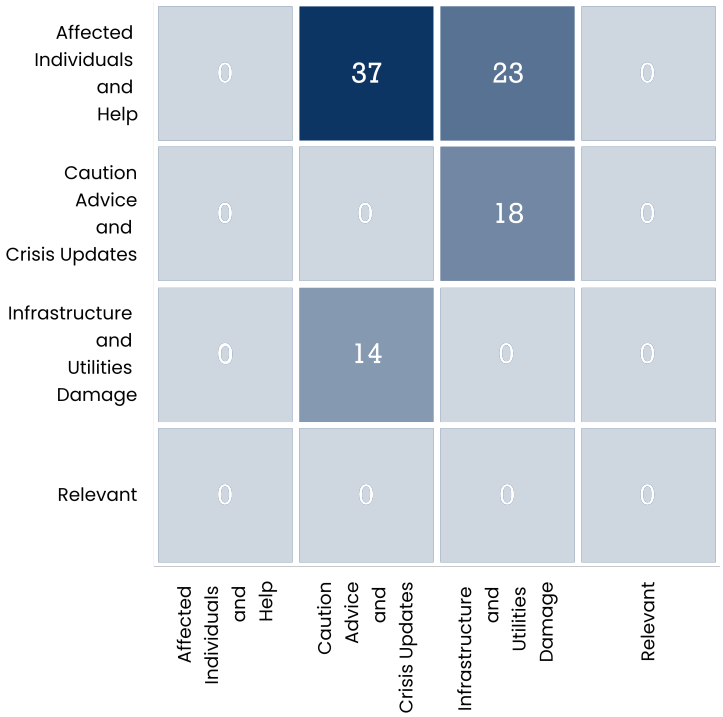

Figure 4.2: Frequency of Class Transitions of Humanitarian Categories

- **Dialect-Inclusive** The dataset includes tweets from various regions across the Arab world, that leads to the diversity of dialects within these tweets. For instance, tweets may originate from countries such as Egypt, Kuwait, Oman, and Lebanon, each characterized by its unique dialect. However, this diversity poses a challenge, as it presents the need for a robust language processing techniques that can handle variations of dialects, ensuring accurate analysis and interpretation of the tweets across different regions.
  The table presented below highlights examples of tweets from different disasters, each originating from a specific location and containing dialect-specific expressions.

Crisis	Tweet's Location	Tweet
Dragon Stroms	Cairo, Egypt	انشروا الارقام ديه فى كل الجروبات لعلها تكون مساعده لأحد من المتضررين من الامطار كان الله فى عون العبد ماكان فى عون اخيه للأبلاغ عن تجمعات مياه الأمطار تليفون 571 الخط الساخن للقاهرة غرفة عمليات محافظة القاهرة 63121932 - 74170932 - 63210932 - 411 منخفض التنين
Kuwait Floods	Kuwait	سيارة اخوي الله ستر عليه اليوم تايرين انشقوا عليه على الدائري السادس مقابل نادي الرماية ! ديروبالكم ! امطار الكويت الكويت تغرق
Coronavirus	Muscat, Oman	صح اني احب الموج وما اصبر من مسقط لكن لو كان شتاء والوضع زي الحين كورونا نو واي اروح ابلي الناس وابلي نفسي ليش لا الموج بيطير ولا صلالة بتطير كلنا فالاخير مكملين ل بعض ومانصبر من بعض لكن للضروره أحكام
Beirut Explosion	Beirut, Lebanon	بيتي صغير بس بساع ألف محب كل من بحاجة لبيت ... بيتي بيتو انفجار المرفأ بيروت

Table 4.1: Dialect Variations In Tweets

4.1.2.2 Text Preprocessing

Text extracted from Social media often contains irrelevant informations such as special characters, user mentions, and URLs. Preprocessing, the process of preparing raw data into a refined format, is where we address this issue. Therefore, we employed the below cleaning to improve the structure of the tweets and minimize noise.

- **Removal of Retweets (RT)** We remove the "RT" (indicating retweets) from the tweet text, as knowing if the tweet has been retweeted or not is not relevant to our tasks.

- **Removal of User Mentions** We removed user mentions, identified by the "@" symbol from the tweet text.

- **Removal of URLs** Any URLs present in the tweet text are eliminated, as they don't significantly contribute additional information, this ensures that they do not influence the analysis.

- **Whitespace Cleanup** Excess whitespaces within the tweet text are removed to ensure consistent formatting.

- **Hashtag Removal**: The "#" symbol, commonly used to denote hashtags in social media posts, is removed from the tweet text but the text is preserved as it often contains relevant keywords or phrases.

4.1.3 Methodology

Our research comprises two distinct tasks aimed at analyzing disaster-related tweets from multiple perspectives.

Classifying the Nature of Disasters

This part consists of classifying the nature of disasters discussed within each tweet, ranging from floods and the COVID-19 pandemic to bombings, explosions, and storms. Each tweet is exclusively associated with one disaster type, making the classification process mutually exclusive.

Classifying Humanitarian Categories

In the second task we aim to classify the humanitarian tweet to the specific message it conveys. This involves identifying different humanitarian concerns, including providing assistance to affected individuals, issuing cautionary advisories and crisis updates, and assessing damage to infrastructure and utilities. Unlike the first task, where each tweet is linked to a single disaster type, in this phase, each tweet can belong to multiple classes. as the tweets often address multiple aspects of humanitarian assistance and response.

4.1.3.1 Voting Classifier

An ensemble learning technique that combines the predictions of multiple classifiers to make a final decision. It operates by aggregating the predictions of its classifiers and selects the class label that have the majority votes, hence the name "voting".

In our approach, we implement soft voting, which takes into account the confidence levels of each classifier's predictions. This allows the voting classifier to make more informed decisions by weighing the certainty of each base classifier to obtain the final prediction results. Below is a figure 4.3 illustrating the architecture of this classifier.

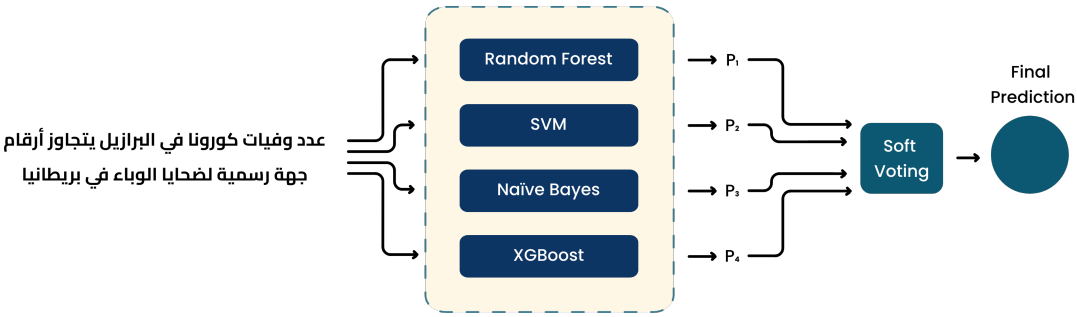

Figure 4.3: The architecture of the Voting classifier

4.1.3.2 mini-BERT

In particular, we used the Asafaya mini-BERT [137], an Arabic pre-trained BERT transformer model Unlike traditional models that process text sequentially, BERT comprehends the contextual nuances of language by considering both preceding and subsequent words simultaneously. This bidirectional approach enables BERT to capture intricate relationships within sentences and extract more nuanced information.

Figure 4.4 illustrates the architecture of BERT, this architecture allows BERT to leverage bidirectional context, significantly enhancing its performance in various NLP tasks.

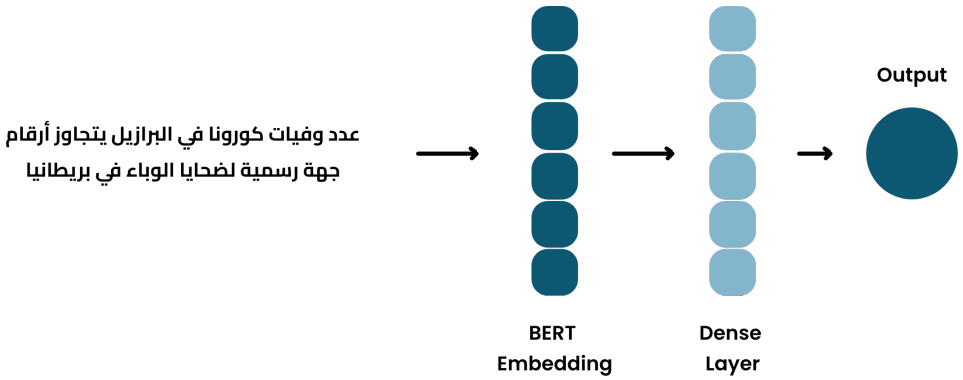

Figure 4.4: The architecture of BERT

4.1.4 Experiments and Results

We conducted several experiments to evaluate the proposed methodologies for disaster type classification and humanitarian needs extraction from social media data.

We trained the traditional machine learning models and we fine-tuned a pre-trained BERT model for both of our classification tasks. The models were trained on a split of the dataset, with 80% used for training and 20% reserved for testing.

4.1.4.1 Disaster's Type

The traditional machine learning models, Random Forest, SVM, NB, and XGBoost, on a per-class basis for disaster type identification. Their performance is detailed in the following table.

Model	**Accuracy per class %**					**Acc. %**
	Bombing	Covid-19	Explosion	Flood	Storm	
RF	**100**	97.83	**100**	**100**	85.71	96.7
SVM	**100**	**100**	**100**	99.64	89.29	97.78
NB	**100**	78.26	**100**	87.05	**100**	93.06
XGB	**100**	97.83	**100**	99.28	**100**	**99.42**

Table 4.2: Traditional Methods Results

Table 4.2 presents the performance of traditional machine learning models on disaster type identification. While all models achieved high accuracy for some classes, the results highlight the potential impact of class imbalance in the data.

Notably, all models achieved perfect accuracy for Bombing and Explosion events. Similarly, SVM excelled at identifying Covid-19, while Random Forest performed exceptionally well in Flood classification, and both NB and XGBoost achieved perfect accuracy for Storm events. These results suggest that the models can effectively classify specific disaster types. However, performance dropped for other classes, particularly for NB with Covid-19. This variation in performance across models and disaster classes underscores the limitations of relying on a single model, especially when dealing with imbalanced data. Here, the concept of an ensemble learning technique like a voting classifier becomes particularly appealing.

To address the limitations of individual models and potentially mitigate the effects of class imbalance, we experimented with a Voting Classifier (VoC). This technique combines the predictions of multiple models, potentially leading to more robust results.

Our exploration involved evaluating various combinations of the four traditional machine learning models presented earlier (Random Forest, SVM, NB, and XGBoost). Through this process, we settled on a voting classifier ensemble comprised of NB and XGBoost as the most effective configuration.

The results of this optimal VoC are presented in the following table.

Table 4.3: Voting Classifier Results

Model	**Accuracy per class %**					**Acc. %**
	Bombing	Covid-19	Explosion	Flood	Storm	
VoC	**100**	97.83	**100**	99.64	**100**	**99.49**

From Table 4.3, the voting classifier using NB and XGBoost achieved an overall accuracy of 99.49%, comparable to the best performing individual model (XGBoost at 99.42%).

We further explored the potential of leveraging advanced neural network architectures. We experimented with asafaya mini-BERT model (mBERT), a pre-trained transformer-based

language model fine-tuned for our disaster type identification task.

Table 4.4: Mini-BERT Results

Model	**Accuracy per class %**					**Acc. %**
	Bombing	Covid-19	Explosion	Flood	Storm	
mBERT	**100**	98.91	**100**	87.05	**100**	**97.19**

As shown in Table 4.4, the mini-BERT model achieved a remarkable overall accuracy of 97.19%. It maintained perfect accuracy for Bombing, Explosion, and Storm classifications, similar to the voting classifier, and a slight improvement in accuracy for Covid-19 classification (98.91%) compared to the voting classifier. However, the mini-BERT model exhibited a performance drop for Flood classification (87.05%) compared to the voting classifier (99.64%). This suggests that while mini-BERT excels at capturing contextual nuances in disaster-related tweets for some classes, further optimization might be necessary to ensure consistent performance across all disaster classes.

Beyond accuracy, computational efficiency is another crucial factor to consider when deploying real-world disaster response systems. In time-sensitive scenarios, rapid classification of disaster types is essential.

Table 4.5 compares the fitting and inference times of the models explored in this study.

Time (s)	**Model**					
	RF	SVM	NB	XGB	VoC	BERT
Fitting	2.72	8.10	**0.04**	10.82	3.54	1133.1
Inference	0.08	0.11	**0.002**	0.01	0.01	21.35

Table 4.5: Fitting and Inference Time

From Table 4.5, the voting classifier offers significant advantages in terms of computational efficiency. The fitting time for the Voting Classifier (3.54 seconds) is comparable to the fastest individual model (NB at 0.04 seconds) and considerably faster than complex models like mini-BERT (1,133 seconds) which is over 300 times slower than the voting classifier.

This efficiency in training time translates to faster inference times as well. As highlighted in Table 4.5, the Voting classifier achieves an inference time of a mere 0.01 seconds for the entire test set. This remarkable efficiency makes it a compelling choice for real-time disaster response applications, where rapid and time-sensitive classification of disaster types is paramount. Conversely, the inference time of mini-BERT stands at 21.35 seconds, representing a significant delay compared to the Voting Classifier precisely 2135 slower than the Voting Classifier. While this delay might appear negligible in non-critical contexts, it can have detrimental consequences

in disaster scenarios. Even a short delay of a few seconds can translate to missed opportunities for timely rescue efforts and aid deployment during critical response windows.

Overall the voting classifier achieved comparable overall accuracy. It surpassed even the advanced mini-BERT model in this regard. However, the true strength of the voting classifier lies in its exceptional efficiency. Its training and inference times are significantly faster than mini-BERT, making it ideal for real-time disaster response applications where speed is paramount.

4.1.4.2 Humanitarian categories

Building upon the previous experiments of traditional machine learning models for crisis type classification, this section examines their performance in a multi-label setting—identifying various humanitarian needs from the same tweets used previously. We focus on four key categories: Affected Individuals and Help (Affected); Caution, Advice & Crisis Updates (Caution); Infrastructure and Utilities Damage (Infra) and Relevant.

Table 4.6 summarizes the results for Random Forest, SVM, NB, and XGBoost. In this table, True Positive indicates the number of tweets correctly classified as belonging to a specific humanitarian need category, while True Negative represents the number of tweets correctly identified as not belonging to that category.

Table 4.6: Traditional Methods Results

Classes		**Models**			
		RF	SVM	NB	XGBoost
Affected Individuals & Help	True Positive	**98.93**	94.10	95.17	95.44
	True Negative	53.12	**77.24**	72.66	65.62
Caution, Advice & Crisis Updates	True Positive	77.17	**85.83**	83.86	76.77
	True Negative	**89.47**	87.04	85.83	85.02
Infrastructure and Utilities Damage	True Positive	**100**	97,78	99.11	98
	True Negative	35.29	**58.82**	47.06	50.98
Relevant	True Positive	**100**	**100**	**100**	**100**
	True Negative	96.74	**100**	92.39	97.83
F1-score		75.63	**82.94**	79.85	78.53

Similar to the crisis type classification, SVM consistently demonstrates strong performance across most humanitarian need categories. It achieves the highest True Positives for Caution, Advice & Crisis Updates (85.83%) and maintains the lead in True Negatives for Affected Individuals and Help (77.24%), Infrastructure Damage (58.82%), and Relevant (100%) for both of the True Positives and the True Negatives.

While SVM excels overall, other models showcase strengths in specific categories. For instance, RF maintains its lead in True Positives for Affected Individuals and Help (98.93%). Additionally, all models achieve perfect True Positives (100%) for Relevant, this might be because tweets classified as Relevant Information appear on their own (without any of the other labels).

Consistent with the crisis type classification, SVM emerges as the top performer with an overall macro F1-score of 82.94%.

As observed in the previous experiments on crisis type classification, the performance of individual models varies across categories. To potentially improve overall performance, we will explore the use of a voting classifier that leverages the strengths of multiple models in the next section. This ensemble approach aims to combine the predictions from various models to arrive at a more reliable final classification for humanitarian needs.

Table 4.7: Voting Classifier Results

	Labels				**F1**
	Affected	Caution	Infra.	Relevant	
TP	92.76	78.74	97.33	**100**	83
TN	**83.59**	**91.50**	**58.82**	**100**	

As detailed in Table 4.7 the voting classifer achieves an F1-Score of 83%, which surpasses the performance of individual models.

In identifying tweets related to Affected Individuals and Help, the voting classifier achieves a True Positives rate of 92.76%, which is comparable to the best individual model (Random Forest at 98.93%). Moreover, the ensemble significantly improves upon the True Negative rate (83.59%) compared to all individual models (ranging from 53.12% to 77.24%). This indicates a reduction in misclassifications for this category. It also demonstrates its strength in identifying Caution/Advice & Crisis Updates as well, achieving a True Positives rate of 78.74%. although this is lower than the best individual model (SVM at 85.83%), the ensemble offers a significant improvement in the True Negatives rate (91.50%) compared to all individual models (ranging from 85.02% to 89.47%). Similar improvements are observed for Infrastructure and Utilities Damage. While Random forest achieved a perfect True Positives 100%, the ensemble maintains that strength with True Positives rate of 97.33% while also boosting the True Negatives rate to 58.82% (the same as SVM's rate). For the last label Relevant it achieves a perfect rate in identifying all the instances.

Like the previous task, we fine-tuned a mini-BERT model (mBERT), its results are in shown in the following table.

Table 4.8: Mini-BERT Results

	Labels				**F1-score**
	Affected	Caution	Infra.	Relevant	
TP	97.32	84.25	**97.78**	**100**	83.90
TN	80.47	**91.90**	54.90	97.83	

Mini-Bert achieves a slightly higher overall Macro F1-score (83.90%) compared to the voting classifier ensemble (83%). It nearly matches the best individual model (Random Forest, 98.93%) in identifying Affected Individuals and Help with 97.32%. Similarly, it takes the lead with the highest True Positives score (97.78%) among all tested approaches for damaged infrastructure & Help, and it even achieves a perfect True Positives score (100%) for Relevant Information.

Beyond accurate identification, mini-BERT exhibits strength in avoiding misclassifications. It boasts a True Negatives rate of 91.90% for Caution/Advice & Crisis Updates, higher than any individual model or even the voting classifier ensemble.

Building on our previous observations about model performance, this table (Table 4.9) sheds light on the computational efficiency of each approach.

Table 4.9: Fitting and Inference Time

Time (s)	**Model**					
	RF	SVM	NB	XGB	VoC	BERT
Fitting	0.98	13.31	**0.1**	23.91	24.57	1070.5
Inference	0.04	0.41	**0.01**	0.05	0.56	14.11

Regarding the fitting time as expected, traditional machine learning models like NB (0.1 seconds) excel in trainig speed. Mini-BERT, on the other hand, requires significantly more time (1070.5 seconds) due to its complex deep learning architecture.

Once trained, all models exhibit speedy prediction times. NB remains the fastest for classifying new tweets (0.01 seconds), followed by the voting classifier ensemble (VoC) at 0.56 seconds. While Mini-BERT takes slightly longer (14.11 seconds).

Mini-BERT's performance is certainly better than all the tested approaches. It achieves a slightly higher Macro F1-score and excels at identifying specific humanitarian needs. However, training Mini-BERT requires considerably more time compared to traditional machine learning models and the voting classifier ensemble. While its inference speed is still suitable for real-time applications, it's not nearly as fast as simpler models, for instance the Mini-Bert model is 25 time slower than the voting classifier.

4.1.5 Traditional ML Insights and the Path to MSA Transfer Learning

The disaster classification study presents several key findings:

- Traditional ML models, despite lacking prior linguistic knowledge, performed remarkably well. This demonstrates that the inherent statistical patterns within Arabic dialectal text are robust enough to be captured by handcrafted features, even in the absence of any pretrained linguistic data.
- Deep learning (BERT), leveraging its pretrained linguistic knowledge, demonstrated strong performance in capturing nuanced patterns in the second stage of experiments. However, in the first stage, its performance was comparable to that of traditional ML models, indicating that while BERT can effectively utilize prior knowledge, it does not always outperform simpler approaches. This highlights potential areas for further improvement.

We are particularly interested in the idea of that simple models, even without any prior linguistic knowledge, can effectively leverage the statistical structure of the data. This indicates that incorporating a related linguistic source—such as MSA—should further enhance performance, particularly for more complex NLP tasks. In other words, the success of traditional ML provides a strong foundation and motivation for exploring transfer learning from MSA, which is linguistically close to Arabic dialects.

Why Transfer Learning from MSA?

The rationale for using MSA-based pretrained models is as follows:

- MSA has more resources compared to Arabic dialects, particularly Algerian Arabic.
- MSA and Arabic dialects share a common linguistic structure, making MSA a natural source for transfer learning.
- Given that models without prior knowledge performed well in our disaster classification task, then models pretrained on a linguistically related source MSA should yield even stronger results, particularly for tasks that require more complex understanding.

Thus, the next phase of our research investigates the impact of MSA-based pretrained models on Arabic dialect identification.

4.2 Leveraging MSA as a Linguistic Bridge for Enhanced Dialect NLP

Dialects, regional variations within a language, characterized by unique pronunciations, grammatical structures, vocabularies, and usage patterns. They act as markers of cultural identity

and social background. Despite the staggering number of languages and dialects spoken globally (over 7,000 according to Ethnologue [69]), a significant disparity exists in the resources available for NLP tasks. This imbalance creates a major challenge: under-resourced languages and dialects lack the resources [101] necessary to develop effective NLP tools, hindering their potential applications in various fields.

This chapter explores a promising solution to bridge this gap: leveraging NLP resources from well-resourced languages to empower the processing of low-resource dialects. This approach aims to transfer knowledge from languages with abundant NLP resources to those with limited resources. By focusing on MSA as a source language, we investigate its potential to enhance NLP capabilities for a specific low-resource dialect —Algerian Arabic. Dialect identification serves as our case study, focusing on the task of automatically pinpointing the dialect used in a text based on its distinct linguistic features.

We compare the performance of three models: AraBERT [37], DziriBERT [1], and mBERT [67], in dialect identification. The results offer valuable insights into developing NLP tools for low-resource dialects and demonstrate the promising potential of using well-resourced languages, including MSA, to support low-resource dialects. This research contributes to the broader goal of enhancing the applicability of NLP tools for a diverse range of languages and dialects to ensure that even under-resourced languages have a voice in the ever-evolving world of NLP.

4.2.1 The MSA versus dialect contrast in Arabic

Arabic, spoken by more than 466 million people globally, is the official or co-official language in around 25 Arabic-speaking countries. It exists in two primary forms, MSA and spoken dialects. MSA, used for formal written communication, official speeches, and taught in schools, is the only standardized and regulated variety. Spoken dialects where Arabic extends beyond a single, standardized form as all Arabic speakers have their own native variety, amounting to approximately 30 different dialects [5]. These dialects introduce additional complexity due to differences in phonology, morphology, lexicon, and syntax [135].

Regional Arabic dialects, primarily used for daily spoken communication, informal conversations, and various forms of popular culture such as music, television, and social media, are neither taught in formal school settings nor commonly used in official written communications. These dialects lack standardized written grammar regulated by a central authority. However, a general understanding of grammatically correct or incorrect usage exists among native speakers, and a sense of shared linguistic norms and conventions has developed over time. Written text can still be produced in these dialects using phonetic spelling rules similar to MSA or adaptations that reflect the dialect's unique characteristics.

Mutual understanding between Arabic dialects is moderate, and an individual's understanding of other dialects depends on their geographical proximity, exposure to cultural works from other countries, and personal experience in interacting with speakers of various dialects. For in-

stance, standard Arabic speakers can typically understand Egyptian Arabic, partly due to Egypt's history in film-making, which made them popular throughout the Arab world. Additionally, the widespread availability of Arabian media, such as television channels and internet platforms, has facilitated the exposure of many Arabic speakers to a broader range of dialects. In contrast, Algerian Arabic is challenging for Arabic speakers from other regions to grasp, especially in its spoken form. This difficulty stems from the unique linguistic features, extensive code-switching, and the influence of Berber, French, and other languages that have shaped the dialect over time.

Dialects can be considered separate languages in their own right, with distinct vocabularies, grammatical structures, and pronunciation patterns that set them apart from MSA and each other. This diversity of dialects within the Arabic-speaking world highlights the importance of accounting for regional variations and linguistic nuances when developing NLP systems or conducting research on Arabic language and dialects.

4.2.2 Enhancing Low-Resource Languages through Well-Resourced ones

There are several strategies for using well-resourced languages to improve low-resource ones in the context of NLP. These approaches include transfer learning, fine-tuning and multilingual models. Transfer learning involves training a model on a well-resourced language task and then applying the learned representations to a low-resource language task. Fine-tuning involves adjusting a pre-trained model on a specific low-resource language task, allowing the model to adapt to the unique characteristics and patterns of the target language. Multilingual models are trained on multiple languages simultaneously, enabling them to learn shared representations and potentially benefit low-resource languages.

Each of these strategies has its strengths and limitations, but they all rely on the promising premise that well-resourced languages can offer valuable information to improve NLP models for low-resource languages. Building on this, this chapter delves into the application of this approach to the task of dialect identification. We specifically explore the use of Well-resourced languages as a source to enhance the performance of NLP models for the Algerian dialect.

4.2.3 Related works

Research on Arabic dialect identification has evolved through the development of specialized corpora and the advancement of pre-trained models. This section first presents key dialectal datasets that have facilitated progress in the field, followed by an overview of pre-trained language models and their role in dialect classification.

4.2.3.1 Arabic Dialect Corpora and Identification

Bouamor et al. [53] introduced the MADAR Arabic Dialect Corpus and Lexicon, a substantial parallel corpus comprising 25 Arabic city dialects in the travel domain and a lexicon of 1,045 concepts with an average of 45 words from 25 cities per concept. Building upon this work, Bouamor et al. [54] organized the MADAR Shared Task on Arabic Fine-Grained Dialect Identification, a pioneering effort that targeted a large set of dialect labels at both city and country levels. This shared task consisted of two subtasks: MADAR Travel Domain Dialect Identification and MADAR Twitter User Dialect Identification.

4.2.3.2 Pre-trained Models

Devlin et al. [67] introduced BERT, a groundbreaking bidirectional encoder representation model from transformers, which has significantly impacted the development of pre-trained models for a wide range of NLP tasks. BERT is designed to pre-train deep bidirectional representations from large-scale unlabeled text data The resulting pre-trained model can be fine-tuned for various tasks, such as question answering and dialect identification, without requiring extensive task-specific architecture modifications.

Following the introduction of BERT, Multilingual BERT (mBERT) emerged as an extension of the original BERT model, designed to handle multiple languages. mBERT was pre-trained on BooksCorpus (800M words) [154] and English Wikipedia (2,500M words), allowing the model to learn shared representations across different languages.

Building upon the foundation laid by mBERT, Antoun et al. [37] introduced AraBERT, a pre-trained transformer-based model specifically tailored for Arabic language understanding. AraBERT highlights the potential of transfer learning and fine-tuning for Arabic language processing, demonstrating its effectiveness in various tasks.

Similarly, Abdaoui et al. [1] presented DziriBERT, a pre-trained language model specifically designed for the Algerian Arabic dialect. DziriBERT showcases the benefits of fine-tuning pre-trained models on dialect-specific data, enabling more accurate representation and understanding of dialect variations within the Arabic language. This approach demonstrates the versatility and adaptability of pre-trained models in handling the linguistic diversity found in Arabic dialects.

4.2.3.3 Leveraging Pre-trained Models for Low-resource Arabic Dialect Identification

Talafha et al. [146] fine-tuned an ArabicBERT to develop a multi-dialect Arabic BERT for country-level dialect identification, demonstrating the utility of pre-trained models in low-resource settings.

Mansour et al. [114] used BERT fine-tuning for Arabic dialect identification in the NADI shared task, showcasing the benefits of transfer learning from well-resourced languages.

Similarly, Beltagy et al. [48] explored Arabic dialect identification using BERT-based domain adaptation, further emphasizing the effectiveness of leveraging pre-trained models to handle low-resource Arabic dialects.

4.2.4 Data description and preprocessing

For training, validating and testing, we used Madar Corpus parallel sentences [53] consisting of parallel sentences translated into the dialects of 25 cities from the Arab World, as well as MSA. For the purposes of our study, we focused specifically on the Algerian Arabic dialect and its corresponding MSA, resulting in a collection of 4000 parallel sentences—2000 in MSA and 2000 in the Algerian dialect.

In the preprocessing stage, we applied several steps to clean and prepare both the Algerian dialect and the MSA data. Firstly, we removed any leading and trailing whitespace characters from each string in the column. We then replaced multiple consecutive whitespace characters with a single space to ensure that each string was properly formatted.

4.2.5 Experiments and results

In this section, we present the evaluation methodology and analyze the performance of the three BERT-based models (AraBERT, DziriBERT, and mBERT) employed for Algerian dialect identification.

To ensure a fair comparison, all three models were trained using the same settings and architecture. We essentially added a new layer to each model specifically designed to classify text samples as either MSA or Algerian dialect.

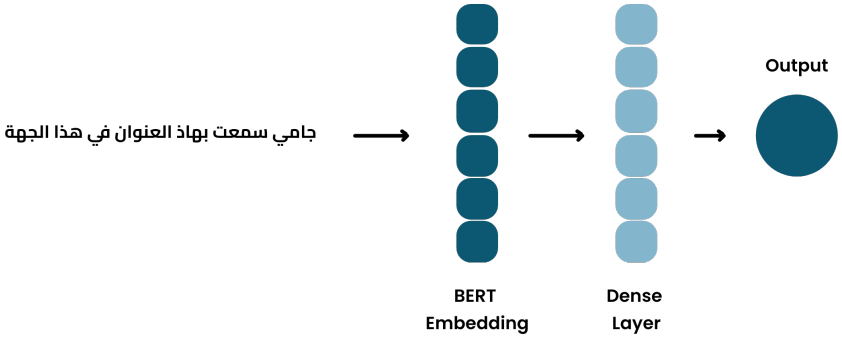

Figure 4.5: BERT-based Models Architecture

The core architecture of the BERT-based models employed in this study is presented in Figure 4.5. This architecture serves as the foundation upon which each model is built, and while each one uses a different pre-trained BERT variant, the overall architecture follows the same

structure with a key component. The Pre-trained BERT Model, depending on the chosen model (AraBERT, DziriBERT, or mBERT), a different pre-trained BERT model is used at this stage.

The performance of each model in classifying Algerian dialects was evaluated using several metrics: accuracy, precision, recall, and F1-score.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
DziriBERT	**96.25%**	**97%**	95%	**96%**
AraBERT	93.38%	90%	**97%**	93%
mBERT	93.62%	93%	94%	94%

Table 4.10: Results of BERT Models

The performance of the three BERT models on the Algerian dialect identification task is summarized in Table 4.10.

DziriBERT achieved the highest accuracy 96.25%. Its precision of 97% indicates a low number of false positives. While its recall is slightly lower at 95%, it still demonstrates robust performance with an F1-score of 96%, reflecting a well-balanced trade-off between precision and recall.

AraBERT achieved an accuracy of 93.38%. However, its precision was lower at 90%, suggesting a higher rate of false positives compared to the other model. This could indicate that AraBERT might occasionally misclassify Algerian dialect samples as Arabic. However, AraBERT compensated for this with the highest recall score of 97%, showcasing its effectiveness in capturing most true Arabic samples. The F1-score of 93% highlights its overall strength, despite the slightly lower precision.

mBERT demonstrates consistent performance across the evaluation criteria, with an accuracy of 93.62%, which is marginally higher than AraBERT but lower than DziriBERT. Its precision of 93% falls between the other two models, better than AraBERT but not as high as DziriBERT. A recall of 94%, and its F1-score of 94% indicates a striking a balance between identifying true Algerian dialect samples and minimizing false positives.

The previous analysis focused on headline metrics (accuracy, precision, recall, F1-score) to provide a high-level overview of model performance. However, a deeper understanding can be gained by examining confusion matrices, which detail how each model classified the data, highlighting both correct and incorrect predictions. Here, we delve into the confusion matrices (Figures 4.6, 4.7, 4.8) for each model to gain a richer perspective on their performance.

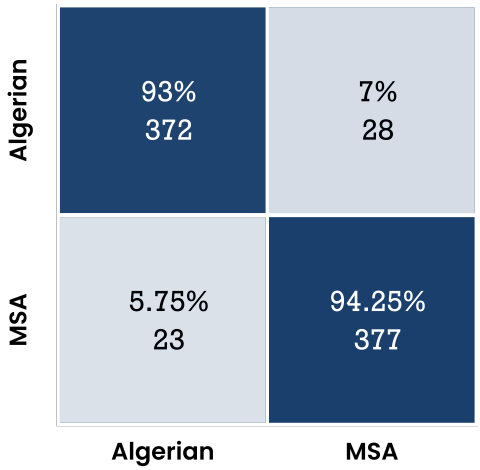

Figure 4.6: mBert confusion matrix

As can be seen, mBERT achieved a high accuracy on both MSA and Algerian dialect samples. It correctly classified 94.25% of MSA samples and 93% of Algerian dialect samples, demonstrating its ability to distinguish between the two. While its overall accuracy is good, the confusion matrix reveals some misclassifications. mBERT misclassified 5.75% of MSA samples as Algerian dialect and 7% of Algerian dialect samples as MSA. These errors highlight areas for potential improvement. The reasons for these misclassifications could be the presence of dialectal variations within the MSA samples or limitations in the model's ability to fully capture certain dialectal features.

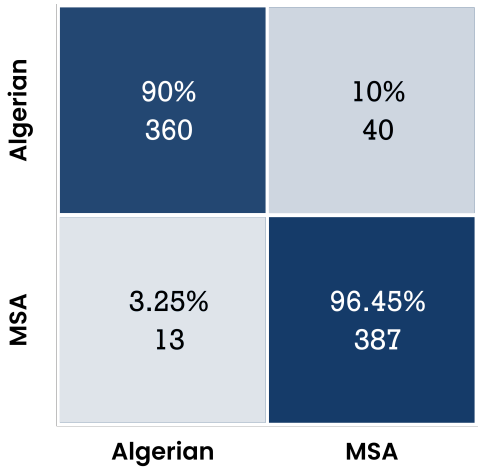

Figure 4.7: AraBERT confusion matrix

AraBERT exhibited a slight improvement over mBERT in MSA classification, correctly identifying 96.75% of MSA samples. This suggests its effectiveness in recognizing standard Arabic text. However, for Algerian dialect samples, AraBERT's accuracy was 90%, indicating a high false positive rate. Further analyzing of the confusion matrix reveals a trade-off in AraBERT's performance. It has a lower false classification rate for MSA samples (3.25%) compared to mBERT,

suggesting better differentiation from Algerian dialect in this category. However, for Algerian dialect samples, the false positive rate is higher (10%) compared to mBERT, indicating AraBERT is missing a larger portion of these samples.

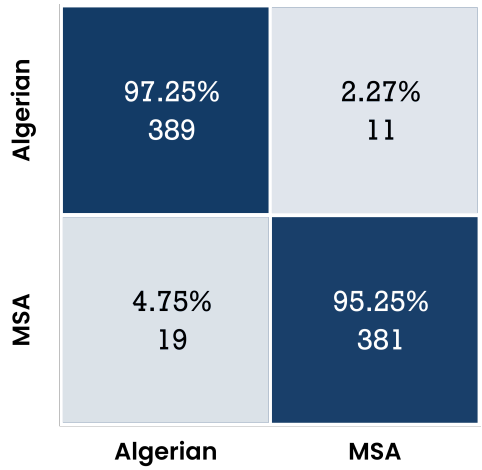

Figure 4.8: DziriBERT confusion matrix

The confusion matrix for DziriBERT solidifies its position as the top performer. It achieved the highest accuracy overall, correctly classifying 95.25% of MSA samples and a remarkable 97.25% of Algerian dialect samples. This demonstrates a strong balance between true positive rates for both MSA and Algerian dialect, suggesting DziriBERT's superior ability to effectively distinguish between the two. Also compared to both mBERT and AraBERT, DziriBERT exhibits the lowest false classification rates. It has the lowest rate for Algerian dialect (2.75%), indicating it excels at identifying these samples correctly. While it has a slightly higher false classification rate for MSA (4.75%) compared to AraBERT, its overall performance remains superior due to its exceptional accuracy in Algerian dialect classification.

The detailed examination of confusion matrices alongside the headline metrics offers a deeper understanding into the strengths and weaknesses of each model. AraBERT excels at identifying MSA text due to its pre-training on a large Arabic corpus. However, its exposure to Algerian dialect variations might be limited, leading to a lower performance in this category. the same goes for DziriBERT, its pre-training on Algerian dialect data gives it a significant advantage in identifying these samples. However, its focus on this specific dialect might result in a slightly lower performance in MSA classification compared to AraBERT. In contrast to these models, which concentrated on a single language during pre-training, mBert is a multilingual model, it achieves a balanced performance, exhibiting some advantages in Algerian dialect classification compared to AraBERT, potentially due to its exposure to various dialects. Furthermore, Algerian dialects often contain a significant amount of French vocabulary and expressions, which could have contributed to mBERT's better performance in Algerian dialect classification, given its multilingual pre-training. However, it is surpassed by DziriBERT, which is specifically tailored

for this task.

Our experiments revealed that AraBERT, DziriBERT, and mBERT models can effectively distinguish Algerian dialect text, with varying degrees of success. DziriBERT achieving the strongest performance followed closely by AraBERT and mBERT. This suggests that models specifically designed for Arabic language understanding tasks or pre-trained on a portion of Arabic text resources are advantageous in arabic dialect identification tasks. These findings support the idea that using pre-trained models on MSA and other well resourced languages holds promise for identifying low-resource dialects like Algerian Arabic.

4.3 Conclusion

The experiments presented in this chapter provide complementary insights into Arabic dialect NLP. On one hand, our disaster classification study showed that traditional ML models—despite lacking any pretrained linguistic knowledge—can effectively capture the inherent statistical patterns of Arabic dialectal text. Their competitive performance, especially in time-critical scenarios, highlights the robustness of handcrafted features and establishes a strong baseline.

On the other hand, our investigation into MSA-based transfer learning revealed that leveraging linguistic knowledge from MSA yields competitive results that closely rival those achieved by specialized model. Although the performance of MSA-based pretrained models was not substantially superior to that of models specifically tailored for the Algerian dialect, the results are highly promising. They indicate that the linguistic proximity between MSA and Algerian dialect can be effectively exploited to enhance model performance. In many instances, the gap between the two approaches was minimal, suggesting that the inherent value of MSA knowledge provides a strong foundation for further improvements.

These findings collectively suggest that if simple models can harness the statistical structure of dialectal text without any prior linguistic information, then integrating MSA—a closely related, resource-rich linguistic source—can further enhance performance in more complex NLP tasks. This is particularly significant given the potential for MSA-based transfer learning to not only offer competitive results but also to enable more scalable and generalizable solutions across various dialect NLP applications. This chapter directly addresses Research Question 1 by evaluating the capabilities of knowledge-free models and demonstrating that their performance supports the hypothesis that MSA-based knowledge can be leveraged to further improve dialect NLP. This chapter addresses the first research question by first evaluating whether models with no prior linguistic knowledge can effectively capture patterns in Arabic dialectal text, and then examining whether MSA-trained models despite having no exposure to dialectal data can generalize to dialectal inputs. The findings support the view that integrating MSA-based knowledge is a promising direction for enhancing performance in low-resource dialectal NLP.

Building on these conclusions, the next phase of our research will focus on the careful design

of models that integrate MSA knowledge. We aim to demonstrate that with refined model architectures and training strategies, it is possible to surpass the performance of specialized models.

The next chapter will detail our efforts to enhance Arabic dialect NLP by fusing MSA-based pretrained representations with innovative model design, aiming to deliver superior results and set a new benchmark in the field.

CHAPTER 5

Cross-Lingual Dialect Identification using Hybrid Architectures: The WASL-DI Approach

This chapter presents the first major contribution of the thesis: the WASL-DI model for dialect identification. It introduces a hybrid architecture that leverages CAMeLBERT and FastText to enhance performance for low-resource dialects like Algerian Arabic. The model is thoroughly evaluated through experiments and ablation study to demonstrate its effectiveness and robustness.

Arabic dialect identification, particularly for low-resource varieties such as Algerian Arabic, presents significant challenges due to limited annotated data, substantial dialectal variation, and the absence of standardized orthography. Arabic is unique in that it sits at the intersection of both high- and low-resource languages. Spoken by over 420 million people worldwide [30]. While MSA benefits from abundant annotated data and established NLP tools [152], Arabic dialects, such as Egyptian, Levantine, and Algerian are often underrepresented, making them low-resource for NLP applications. Despite MSA's unifying role and shared features with dialects, the significant variations in vocabulary, grammar, and pronunciation make it difficult to develop accurate, generalizable dialect identification models. This challenge has led to the creation of various datasets such as MADAR [53], NADI [7–11], QADI [4], and ADI17 [141], which aim to facilitate research in this domain.

Building on our previous findings, which demonstrated that simple, handcrafted models can capture robust statistical patterns in dialectal text, and that leveraging related linguistic knowledge from MSA can yield competitive performance this chapter introduces our novel hybrid cross-lingual model. This model is designed to address the data scarcity and linguistic diversity inherent in Arabic dialects while exploring how MSA knowledge transfer can enhance dialect identification for low-resource varieties like Algerian Arabic.

Our approach fuses the deep contextual representations provided by CAMeLBERT [91], a transformer-based model pretrained on large Arabic corpora, with the subword-level semantic richness of FastText embeddings [52]. By combining these two complementary sources, the model is designed to capture both high-level linguistic features and fine-grained statistical patterns inherent in dialectal Arabic. This hybrid architecture not only leverages the strengths of pretrained models but also compensates for the lack of large dialect-specific datasets. While BERT-based models [67] capture dynamic, context-aware representations, FastText embeddings are particularly useful for handling OOV words and modeling subword information, both of which are critical for dialect identification due to the morphological complexity of Arabic dialects.

This hybrid design is motivated by the observation that while specialized models perform well when trained on dialect-specific data, MSA-based models offer competitive results, even without fine-tuning on specific dialects. This insight suggests that by carefully combining MSA knowledge with adaptable feature extraction techniques, we can develop a model that not only matches but potentially surpasses the performance of specialized models—especially when annotated dialectal data is limited.

In this chapter, we detail the design, implementation, and evaluation of our proposed hybrid model. We describe the model architecture, including how the dual representations are integrated and optimized to capture both local and contextual dialectal features. To evaluate the model's effectiveness, we conduct extensive experiments on MADAR and other datasets.

We also perform a comprehensive ablation study to evaluate the contribution of each model component, offering insights into how the fusion of MSA knowledge and dialectal information impacts performance. Our results demonstrate that this hybrid model not only delivers competitive performance compared to specialized models but, with careful architectural design, can outperform them across several dialectal tasks.

This contribution advances the state of the art in Arabic dialect identification by addressing the core challenges of low-resource NLP through a hybrid, cross-lingual approach. Specifically, this work:

1. Mitigates data scarcity by combining FastText subword representations with BERT's context-aware embeddings, ensuring better handling of OOV words and capturing both broad semantic and dialect-specific linguistic nuances.

2. Leverages MSA resources to transfer linguistic knowledge to Arabic dialects, improving generalization capabilities with minimal dialect-specific data.

3. Evaluates the hybrid approach on diverse datasets, rigorously analyzing how different model components contribute to overall performance and assessing the model's adaptability across a range of dialectal contexts.

4. Demonstrates significant improvements over prior approaches, establishing a new benchmark for low-resource dialect identification and providing insights for future cross-lingual NLP methodologies.

Through this work, we aim to establish a scalable and adaptable framework for handling dialects with limited resources while also providing a model architecture that can be extended to other underrepresented languages and dialects. This chapter lays the foundation for the next phase of research—exploring how careful model design and the strategic use of MSA knowledge can deliver superior results in Arabic dialect identification.

Publication Note

The work presented in this chapter has resulted in one journal publication

- M. Chabane, F. Harrag, and K. Shaalan, "Advancing low-resource dialect identification: A hybrid cross-lingual model leveraging CAMeLBERT and FastText for Algerian Arabic," *Expert Systems with Applications*, vol. 284, 2025, Art. no. 127816, doi: 10.1016/j.eswa.2025.127816.

5.1 Related Works

ADI has been the focus of several prior studies, as detailed in the related work section (see Chapter 3 specifically Section 3.1). These works have introduced key datasets and methodologies that serve as a foundation for our approach. Among the most influential resources are the MADAR corpus [53], which provides fine-grained dialectal data spanning 26 Arab cities, and the NADI shared task series [7–11], which evaluates dialect identification across various Arabic varieties. Additionally, the QADI [4] and ADI17 [141] datasets offer valuable benchmarks for developing and comparing dialect classification models.

While these studies have advanced the field through dataset creation and benchmark evaluations, most existing approaches rely on either traditional ML models or specialized ML architectures. Such approaches, while effective when substantial training data is available, struggle with generalization to underrepresented dialects due to the inherent data scarcity. Moreover, few studies have explored leveraging MSA knowledge to improve performance on low-resource dialects.

Our work addresses these limitations by proposing a hybrid model that combines the strengths of both CAMeLBERT and FastText embeddings. This approach is informed by prior findings on the utility of subword representations and contextual embeddings in dialectal NLP. By incorporating knowledge transfer from MSA and fusing dual representations, our model not only improves dialect identification for low-resource varieties like Algerian Arabic but also provides a scalable framework adaptable to other underrepresented languages. This extends prior research

by offering a comprehensive evaluation of how cross-lingual knowledge transfer and hybrid architectures can mitigate the challenges posed by dialectal variation and limited annotated resources.

5.2 Methodology

In this section, we describe the approaches and models we used to address the task of dialect identification, ranging from traditional deep learning methods to state-of-the-art BERT-based models and our proposed hybrid approach. We detail the model architectures, hyperparameters, and the training procedures followed to ensure consistency and robustness in our evaluation.

5.2.1 Hyperparameters Summary

To ensure consistency and fairness, all models were trained using the same set of hyperparameters. Additionally, each model was run using five different random seeds to account for variance in performance and ensure the robustness of the results. The use of multiple seeds helps mitigate the effect of random initialization, providing a more reliable evaluation of model performance. Table 5.1 summarizes the key hyperparameters used for traditional deep learning models, BERT-based models, and the proposed hybrid approach.

Parameters	Value
Epochs	20
Batch size	32
Learning rate	1×10^{-5}
Optimizer	Adam
Cost function	BinaryCrossentropy()
Activation function (Hidden layer)	Relu
Activation function (Output layer)	Sigmoid
Embedding Dimension (Fasttext Embedding)	150
Units in BiLSTM / BiGRU	384
CNN Filters	384

Table 5.1: The hyperparameters of our models

5.2.2 Baselines

To ensure a comprehensive comparison, we implemented several baseline models. These baselines include traditional deep learning models and BERT-based architectures, which serve as reference points for evaluating the effectiveness of our proposed hybrid model.

5.2.2.1 Traditional Deep Learning Models

In this study, traditional deep learning models, including CNN, BiLSTM, BiGRU, CNN-BiGRU, and CNN-BiLSTM, were implemented to serve as baselines for our dialect identification task. To ensure a fair and unbiased performance comparison, we employed a shared embedding layer, a pre-trained FastText embeddings, specifically designed for Arabic text, across all models to maintain consistency in input representations.

The architecture of these models follows a common structure, with variations introduced in the *Variable Layer*, as illustrated in Figure 5.1. The *Variable Layer* is where the core functionality of each model diverges, depending on whether CNN, BiLSTM, BiGRU, CNN-BiGRU, or CNN-BiLSTM is employed.

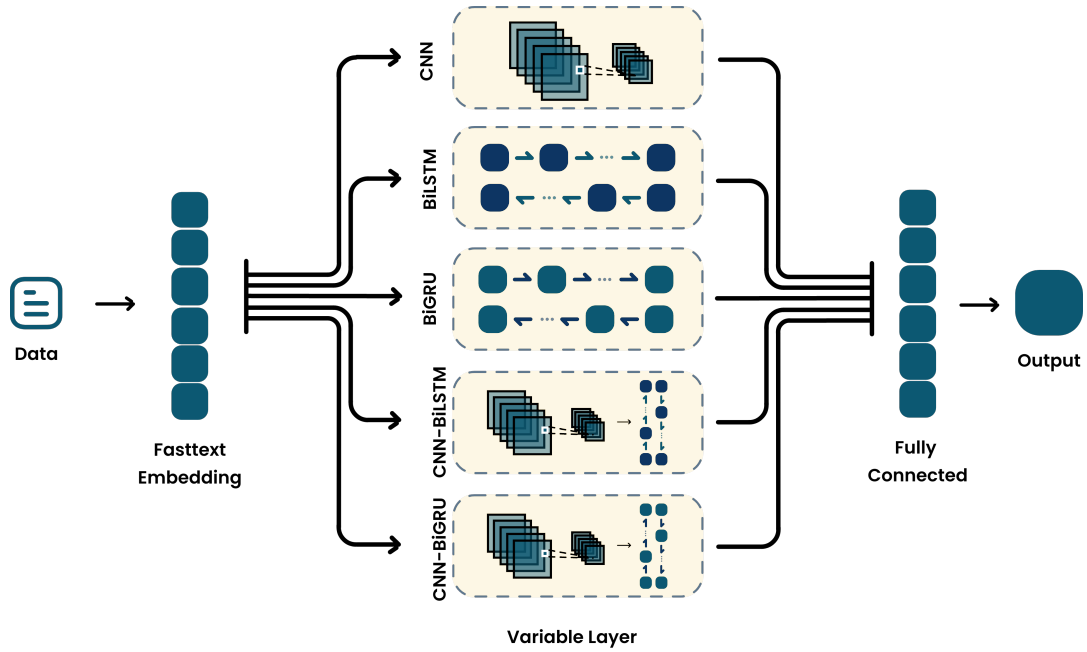

Figure 5.1: The architecture of traditional deep learning models used in this study for dialect identification.

- CNN applies a 1-dimensional convolutional layer with 32 filters and a kernel size of 3. It is followed by a Rectified Linear Unit (ReLU) activation function. The CNN focuses on capturing local patterns in short sequences, such as word n-grams, and employs global max pooling to downsample the feature maps, reducing dimensionality and focusing on the most important features.

- The BiLSTM [89] model uses 32 units to capture long-range dependencies in the text by processing the input sequence in both forward and backward directions. The bidirectional nature allows the model to better understand context by taking into account the entire sequence, improving its ability to capture relationships between distant words.

- The BiGRU shares a similar architecture with BiLSTM, but uses GRU cells instead. GRUs are known for their efficiency in training due to their simpler gating mechanism, making them suitable for sequence modeling tasks where computational resources may be limited. Like BiLSTM, the BiGRU processes text in both directions.

- The CNN-BiGRU (Convolutional-BiGRU Hybrid) combines CNN and BiGRU models into a single framework. The CNN component captures local patterns in the input text, while the BiGRU processes these patterns sequentially to capture long-range dependencies. The hybrid nature of this model allows it to benefit from both local and global contextual information.

- The CNN-BiLSTM (Convolutional-BiLSTM Hybrid) is similar to CNN-BiGRU, but combines a CNN with a BiLSTM layer. The convolutional layer captures local features, while the BiLSTM layer ensures that long-range dependencies are modeled effectively. This combination provides a powerful approach to handling both local and global aspects of the text.

- Each model includes a *Fully Connected Layer* with 1024 neurons and a ReLU activation function, which learns complex patterns from the extracted features. This is followed by an *Output Layer* with one neuron and a sigmoid activation function, which outputs the final classification probabilities.

5.2.2.2 BERT-based Models

In this experimental phase, we used pre-trained language models based on the BERT architecture, including DziriBERT [1] and MDABERT [147]. These models, specifically designed for processing Arabic dialects, were fine-tuned on our binary classification task. The fine-tuning involved adding a dense output layer to the pre-trained encoder, allowing the model to map encoded representations to binary class labels. Both models benefit from extensive pre-training on Arabic text, with DziriBERT focusing on Algerian Arabic and MDABERT on MSA and DA.

The two models were fine-tuned by introducing a classification head with a sigmoid function. Figure 5.2 illustrates the architecture of our BERT-based models.

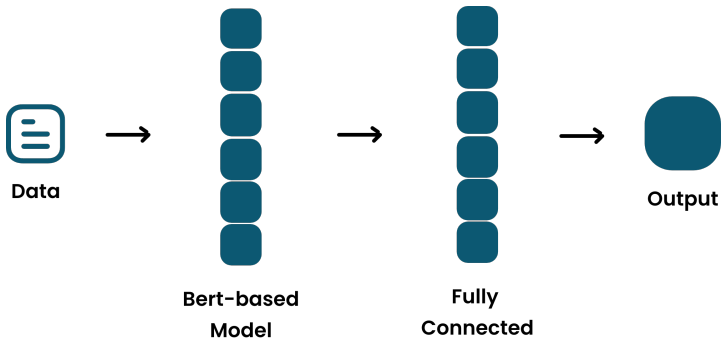

Figure 5.2: Bert-based Models Architecture

5.2.3 Proposed Approach

In this section, we describe the proposed hybrid model, WASL-DI (Wielding Arabic resources to Support Low-resource Dialect Identification), designed for our dialect identification task. The model leverages the strengths of both pretrained language models (PLMs) and deep neural networks (DNNs) to effectively capture both contextual and local linguistic features.

Figure 5.3 illustrates the architecture of our proposed hybrid model.

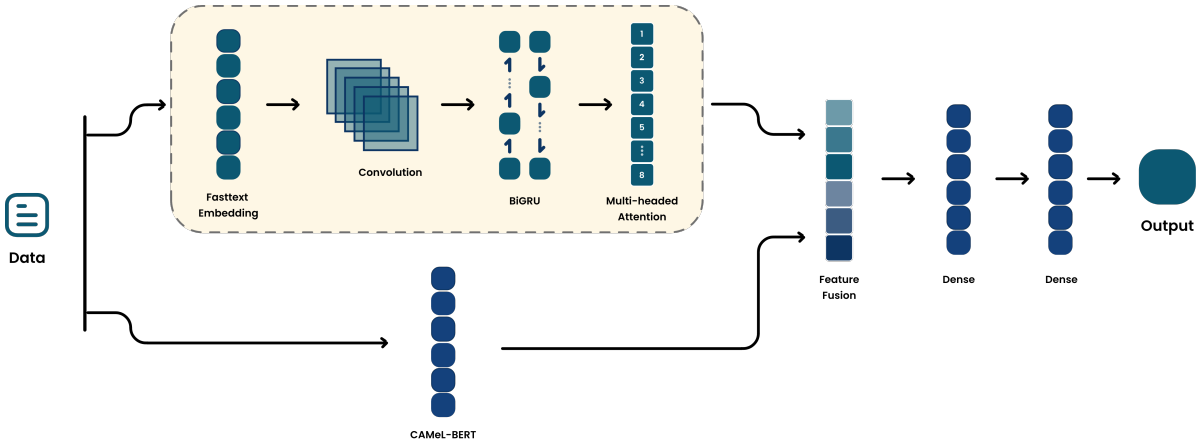

Figure 5.3: Proposed Hybrid Model Architecture

5.2.3.1 Detailed Architecture

The hybrid model processes the input text through two parallel paths. On the first path, the raw text is transformed into numerical representations using FastText embeddings, which are designed to capture local contextual information. Simultaneously, on the second path, CAMeL BERT, a pre-trained transformer model on MSA, is used as the contextual encoder. CAMeL BERT tokenizes the input sequences and generates hidden state representations for each token, which are then summarized via global average pooling to produce a fixed-size vector encapsulating the meaning of the entire sentence.

While the FastText embeddings are passed through a CNN, extracting local features from the text, the BiGRU layer then processes the CNN's output. The BiGRU captures long-range dependencies in both forward and backward directions, ensuring that the context of each word is considered across the entire sentence. We chose BiGRU over BiLSTM due to its simpler design and lower computational cost, this allows us to maintain high accuracy while improving efficiency.

For each token X_i^t, the forward GRU processes the sequence from left to right, updating the hidden state $\overrightarrow{H_{i,\text{GRU}}}$ based on the current token and the hidden state from the previous token:

$$\overrightarrow{H_{i,\text{GRU}}} = \text{GRU}(X_i^t, \overrightarrow{H_{i-1,\text{GRU}}}) \quad \text{for } i = 1,2,\ldots,n \tag{5.1}$$

For the backward GRU, i varies from n down to 1, This means i represents the current token being processed from right to left.

$$\overleftarrow{H_{i,\text{GRU}}} = \text{GRU}(X_{t_i}, \overleftarrow{H_{i+1,\text{GRU}}}) \quad \text{for } i = n, n-1,\ldots,1 \tag{5.2}$$

The overall hidden state H_{GRU} combines the forward and backward hidden states :

$$H_{\text{GRU}} = [\overrightarrow{H_{\text{GRU}}}, \overleftarrow{H_{\text{GRU}}}] \tag{5.3}$$

The output from the BiGRU is then passed through a Multi-Head Attention mechanism [150]. This layer allows the model to focus on different parts of the sequence and assign varying importance to specific words or phrases, depending on their relevance within the sentence.

In Multi-Head Attention, we have h heads, and the output is concatenated:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W_O \tag{5.4}$$

Where each head is defined as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{5.5}$$

Here:

- W_i^Q, W_i^K, W_i^V are all learnable parameter matrices.

The outputs from CAMeL BERT, after applying global average pooling, are merged with the output of the CNN-GRU-attention pipeline, which also undergoes global average pooling. This fusion layer combines the deep contextual embeddings from CAMeL BERT with the localized and sequential features captured by the CNN-GRU-attention layers, resulting in a richer and more comprehensive representation of the input.

The fused features are then fed into a Multi-Layer Perceptron, which consists of several fully connected layers. The MLP helps in learning higher-order interactions between the fused

features to refine the representation. The final output layer is a sigmoid-activated neuron that produces a probability score for the input belonging to a specific dialect class. If the probability exceeds the predefined threshold of 0.5, the model classifies the text as belonging to the positive class, otherwise, it assigns it to the negative class.

To improve generalization and stability during training, batch normalization is applied throughout the model, and dropout is used to prevent overfitting.

5.3 Experimentation and Results

In this section, we detail our experimental framework, which encompasses the datasets used for training and evaluation, the results and discussion, validation processes, an exploration of the generalizability of WASL-DI, and an examination of its limitations.

5.3.1 Dataset

Due to the scarcity of publicly available Algerian dialect data, we adopted a multi-source approach to construct a representative corpus for this under-resourced language. We formed two datasets one for training, validation and testing, the other one for model's external validation.

5.3.1.1 Training, validation & testing dataset

For training, validation, and testing purposes, we employed six diverse datasets. To construct the MSA sentence corpus, we used the 100k Arabic reviews dataset [107], a compilation of various publicly accessible datasets, sampled to encompass 100,000 instances. This dataset contains reviews from hotels, books, movies, products, and a subset of airline reviews. Originally classified into three categories (Mixed, Negative, and Positive), these original sentiment labels were discarded for our study, and the entire dataset was uniformly assigned the MSA label. To achieve parity with the Algerian dialect dataset, we extracted a balanced subset of exactly 82,228 samples from the corpus.

As for the Algerian dialect sentences, five distinct datasets were incorporated comprising 82,228 sentences that were all labeled Algerian 5.2.

Dataset	Source	Size	Selected Subset
Algerian Car Market Sentiment	[55]	27.4k	Entire
Algerian Dialect	[50]	45k	Entire
Algerian Corpus	[94]	6.8k	Entire
Madar CORPUS-25 (Algerian)	[53]	52k	2k sentences
NADI TWT-2023 (Algerian)	[7]	23.4k	1k sentences

Table 5.2: Overview of Algerian Datasets Used for Analysis, Including Source References, Total Size, and Selected Subsets for Model Training and Evaluation.

To better understand the composition of our datasets, the following figure illustrates the vocabulary size and the number of sentences for each language class.

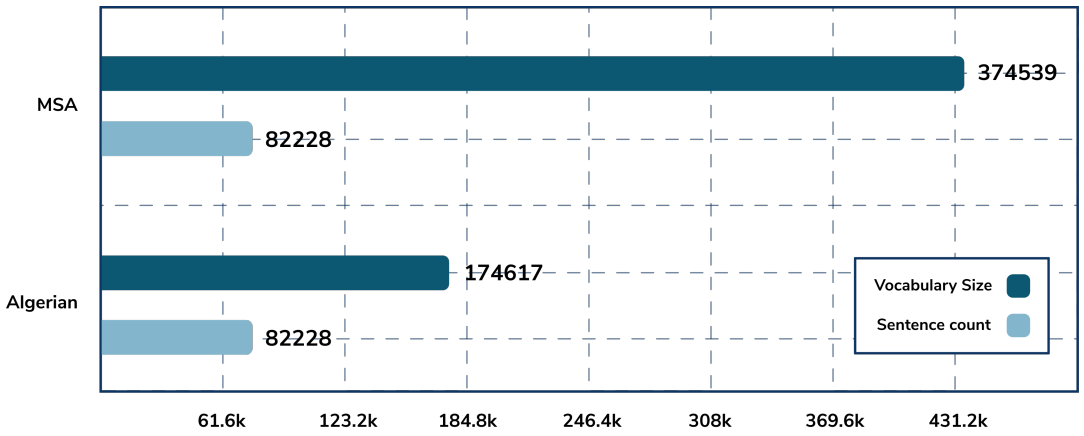

Figure 5.4: Comparison of vocabulary size and sentence count between MSA and the Algerian dialect.

To understand the linguistic disparities between MSA and Algerian Arabic, we analyzed vocabulary size and sentence count. As illustrated in Figure 5.4, both datasets comprise approximately 82.2k sentences. However, a significant difference emerges in vocabulary size: MSA exhibits a substantially larger lexicon (374,539 unique words) compared to Algerian Arabic (174,617 unique words). This disparity is not attributable to data quantity but rather reflects inherent lexical differences between the two language varieties. MSA's expansive vocabulary suggests a broader lexical range and potential complexities for natural language processing tasks, while Algerian Arabic's more concentrated lexicon may indicate a specific lexical focus. The contrast in vocabulary size, despite equal sentence counts, underscores the varying lexical richness and representational challenges between the two datasets.

5.3.1.2 External validation dataset

To ensure a comprehensive evaluation of WASL-DI performance and generalizability, we constructed a distinct validation dataset that was entirely independent from the training data. This external validation dataset comprised approximately 12k sentences in each one of the following dialects, Moroccan, Tunisian, and MSA, sourced from the Madar corpus. Additionally, we included another 12k sentences from Algerian-Darija [109], a specific dialect of Algerian Arabic, to further assess the model's ability to handle diverse regional variations.

By using this external validation dataset, we were able to evaluate the model's performance on another data, providing insights into its ability to generalize and handle linguistic variations beyond the Algerian dialect.

5.3.2 Statistical Tests

Both McNemar's test and the paired t-test were employed to rigorously assess the significance of performance improvements in our ablation study, where we evaluated the contribution of each component in the proposed hybrid model. All tests were conducted at a stringent significance level of $\alpha = 0.02$, ensuring the reliability of our results and reducing the likelihood of Type I errors.

McNemar's Test: McNemar's test [116] is applied to compare the performance of two classifiers on the same set of instances. It is particularly effective for binary classification tasks, as it focuses on the disagreements between the classifiers. The test constructs a 2×2 contingency table, summarizing the predictions from both classifiers:

	Model 2 Correct	Model 2 Incorrect
Model 1 Correct	n_{11}	n_{12}
Model 1 Incorrect	n_{21}	n_{22}

Table 5.3: Contingency table for McNemar's test

Here, n_{12} is the number of instances where Model 1 is correct and Model 2 is incorrect, and n_{21} is the reverse. The test focuses on these discordant pairs. The McNemar test statistic is calculated as:

$$\chi^2 = \frac{(n_{12} - n_{21})^2}{n_{12} + n_{21}} \tag{5.6}$$

To determine the significance of the results, we calculate the p-value, which represents the probability of obtaining the observed test statistic (or one more extreme) under the null hypothesis. The null hypothesis assumes that both models perform equally well, i.e., the probabilities of making errors are the same for both classifiers.

- If the p-value is less than the significance level ($\alpha = 0.02$ in our case), we reject the null hypothesis, concluding that there is a statistically significant difference between the two models.
- If the p-value is greater than α, we fail to reject the null hypothesis, implying that any observed difference in performance could be due to random chance.

Paired t-Test: In addition to McNemar's test, we used the paired t-test to compare the means of two related samples (the accuracies of the five runs across each seed). The paired t-test measures whether the average difference between paired observations is significantly different from zero.

The test statistic is computed as follows:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} \tag{5.7}$$

Where:

- $\bar{d}$ is the mean of the differences $d_i = X_{1i} - X_{2i}$,
- s_d is the standard deviation of the differences,
- n is the number of paired observations.

The resulting p-value indicates whether the observed mean difference is statistically significant. Similarly to McNemar, If the p-value is less than $\alpha = 0.001$, we reject the null hypothesis, concluding that the difference in performance between the models is statistically significant. If the p-value exceeds α, we fail to reject the null hypothesis, meaning that any observed differences in means could be due to chance.

The combination of both statistical tests in the ablation study provided robust evidence that each modification in the model architecture contributed meaningfully to the overall performance, and the improvements observed were not due to chance.

5.3.3 Results and Discussion

In this section, we present the outcomes of our experiments, focusing on a thorough assessment of each model's ability to differentiate between Arabic and Algerian dialect. Our analysis encompasses a diverse range of deep learning models, from traditional architectures like CNN and BiLSTM, to more sophisticated structures such as hybrid models and BERT-based methodologies.

Category	Model	Accuracy%	Arabic Acc.%	Algerian Acc.%
Traditional Machine Learning Models	CNN	94.55 (±0.25)	92.04 (±0.78)	97.08 (±0.31)
	BiLSTM	96.81 (±0.3)	95.34 (±1.55)	97.84 (±0.95)
	BiGRU	95.81 (±0.3)	96.31 (±1.54)	96.32 (1.57)
	CBiGRU	95.55 (±0.3)	94.73 (±1.77)	96.38 (±2.23)
	CBiLSTM	95.63 (±0.23)	93.56 (±0.86)	97.71 (±0.52)
BERT-based Models	Dziri-BERT	98.92 (±0.12)	98.09 (±0.28)	**99.74 (±0.05)**
	MDA-BERT	98.96 (±0.08)	98.33 (±0.21)	99.59 (±0.04)
Our Model	WASL-DI	**99.24 (±0.02)**	**99.01 (±0.21)**	99.46 (±0.09)

Table 5.4: Performance Results of Various Models for Dialect Identification, Including Accuracy Metrics for Overall Performance, Arabic-Specific Accuracy and Algerian-Specific Accuracy. The values reported are the mean accuracy across five runs with different random seeds, along with the standard deviation.

The results presented in Table 5.4 demonstrate the performance of baseline models and our proposed approach. In the category of traditional machine learning models, the BiLSTM model achieves the highest accuracy of 96.81% (±0.30), closely followed by the BiGRU and CBiLSTM models. While these models are effective, they may not fully capture the intricate linguistic features of dialects.

BERT-based models exhibit a significant improvement in accuracy, with Dziri-BERT achieving 98.92% (±0.12) and MDA-BERT at 98.96% (±0.08). As expected, Dziri-BERT performs best on Algerian data (99.74%), given that it is specifically trained on this dialect, whereas MDA-BERT's performance across dialects reflects the benefits of training on broader dialectal data.

Notably, our proposed hybrid model not only surpasses both Dziri-BERT and MDA-BERT in overall accuracy, achieving 99.24% (±0.02), but also excels in handling both Arabic and Algerian categories. Although it does not outperform Dziri-BERT on Algerian data, it remains competitive (99.46%), ensuring that it does not lag significantly behind. More importantly, it achieves the highest performance on Arabic data (99.01%), contributing to its superior overall accuracy.

This balance between Arabic and Algerian accuracy indicates that the integration of CAMeL BERT and FastText embeddings allows for effective transfer of MSA resources to dialectal contexts without sacrificing performance in either category. The strong results across both languages validate our assumption that MSA resources can be successfully leveraged to enhance dialect identification.

5.3.4 Validation

To validate WASL-DI, we performed a 5-fold cross-validation technique. This approach systematically divides the available data into training and validation sets, and this procedure is carried out for each fold. Ensuring every data point is used for both training and validation, and the average performance across all folds is considered the final metric.

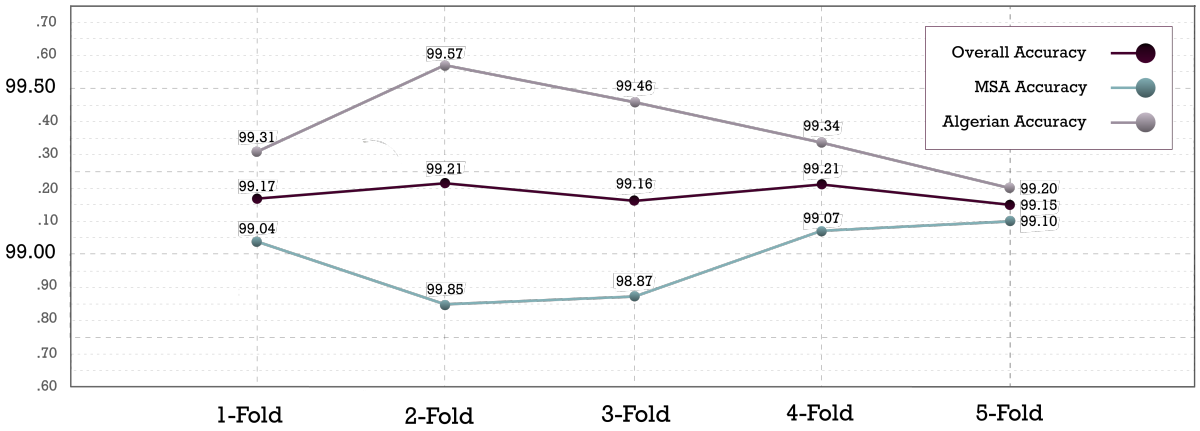

Figure 5.5: 5-Fold Cross-Validation Results

The cross-validation results demonstrate the consistent performance of WASL-DI across

different data partitions. The overall accuracy, MSA accuracy, and Algerian accuracy remained relatively stable throughout the five folds, indicating that the model is not overly sensitive to the specific data split and is capable of achieving high accuracy across different data partitions.

5.3.5 Robustness to Noisy and Incomplete Data

To evaluate the robustness of WASL-DI in real-world scenarios where data is often noisy or incomplete, we conducted experiments by systematically introducing noise and incompleteness into our dataset. These experiments simulate the challenges commonly encountered in real-world applications, such as social media text or user-generated content, where noise (e.g., typos, misspellings) and incompleteness (e.g., truncated sentences, missing words) are common. By testing the model under these conditions, we aim to assess its ability to maintain high performance in practical, imperfect environments.

5.3.5.1 Noisy Data Experiment

We introduced noise into our dataset using a custom noise function designed to simulate typos and other common errors. This function randomly applies one of the following operations to each input text:

- **Character Replacement:** Randomly replaces characters with other ones (simulating typos).
- **Character Insertion:** Randomly inserts characters into the text (simulating extra keystrokes).
- **Character Deletion:** Randomly deletes characters from the text (simulating missing keystrokes).
- **Shuffling:** Shuffles words or characters within the text (simulating disorganized input).
- **No Operation:** Leaves the text unchanged (to simulate cases where no noise is present).

The noise level was set to 10%, meaning each character had a 10% chance of being affected by one of the noise operations. This level of noise simulates moderate to severe degradation of the input data, including realistic typos and errors.

Examples of noisy sentences are shown in Table 5.5.

Original Sentence	Noisy Sentence
صح خويا محمد ربي يخليك خاصني شي حاجه شي حاجه مادباسيش 0053 ربي بخليك	صح ژخويا محمد غربِيk ک❁يخل٦يك خايصِني شي حاجهه شي حاجه وتمادباسيش 0ً053 ربي بخليك
القصة رائعة و لكن النهاية مش مقنعة تماما	القصة رئعة و لكن النهاية مش لقنعة تماگا
Grand escro w criminal	w criminal escro Grand ....

Table 5.5: Examples of noisy sentences

To evaluate the impact of noise on WASL-DI's performance, we compared its accuracy on clean data versus noisy data. The results, summarized in Table 5.6, demonstrate the model's ability to maintain high accuracy even when exposed to significant noise.

Data	Accuracy%	MSA Accuracy%	Algerian Accuracy%
Clean data	99.24	99.01	99.46
Noisy data	96.16	95.58	96.74

Table 5.6: Performance of WASL-DI on noisy data.

Despite the introduction of significant noise, including typos, across the entire dataset, WASL-DI maintained a high accuracy of 96.16%, demonstrating its robustness to noisy inputs. The minor drop in performance (3.08%) suggests that the model can effectively handle real-world scenarios where data quality may vary.

5.3.5.2 Incomplete Data Experiment

To simulate incomplete data, we randomly removed words from all the data. The incompleteness level was randomly chosen between 10% and 50% for each text, meaning that between 10% and 50% of the words in each sentence were removed. This simulates cases where parts of the text are missing or truncated, such as fragmented user inputs. Examples of incomplete sentences are shown in Table 5.7.

Table 5.7: Examples of incomplete sentences

Original Sentence	Incomplete Sentence
اليوم، نروح نشوف كيفاش دايرة شايناطاون.	اليوم، نروح كيفاش دايرة
مرضي. الموقع جدا رائع وجميل قربه من الحرم. الأثاث قديم نوعا ما	مرضي. الموقع جدا وجميل قربه الأثاث نوعا
trop chère a cause de notre dinars	cause dinars

To measure WASL-DI's ability to handle incomplete data, we tested its performance on the incomplete data The results are shown in Table 5.8.

Data	Accuracy%	MSA Accuracy%	Algerian Accuracy%
Clean data	99.24	99.01	99.46
Incomplete data	96.90	96.76	96.63

Table 5.8: Performance of WASL-DI on incomplete data.

The model achieved an accuracy of 96.90% on incomplete data, with only a 2.34% performance drop compared to its performance on complete data. This demonstrates that WASL-DI is highly robust to missing or truncated text.

The results underscore WASL-DI's reliability in practical use cases, such as text processing in noisy environments or applications with imperfect data sources. However, the minor performance drop observed in these experiments suggests that there is room for improvement, particularly in handling more extreme cases of data incompleteness and noise.

5.3.6 Generalizability of WASL-DI

To demonstrate the language independence of WASL-DI, we focused on its ability to generalize across different Arabic dialects and data scenarios. The goal is to determine if a model pre-trained solely on MSA could perform well across a diverse range of Arabic dialects.

5.3.6.1 Magharebi Dialect

We conducted a validation using a dataset of 1k sentences, with 250 sentences each from MSA, Algerian, Moroccan, and Tunisian dialects. The training dataset was deliberately kept small, 250 sentences per dialect for training, 40 per dialect for validation, while the the remaining data, comprising approximately 11.7k sentences per class, was used for testing. This setup was intentionally designed to test the model's ability to generalize from a minimal amount of training data to a much larger and more diverse test set.

We evaluated WASL-DI, against DziriBERT, DarijaBERT, and MDA BERT, which were pretrained on Algerian, Moroccan dialects, and diverse Arabic dialects, respectively. these models were fine-tuned on the training set. The comparison results between them are shown in Table 5.9.

Model	Accuracy	Algerian Accuracy	Morrocan Accuracy	MSA Accuracy	Tunisian Accuracy
DziriBERT	71.72	98.21	52.60	70.67	65.35
Darija BERT	70.31	94.66	64.13	66.33	56.04
MDA BERT	60.29	88.98	43.41	63.72	44.94
WASL-DI	**92.13**	**99.90**	**87.52**	**98.19**	**82.83**

Table 5.9: Accuracy Comparison of DziriBERT, DarijaBERT, MDA BERT, and WASL-DI Across Modern Standard Arabic (MSA) and Three Maghrebi Dialects (Algerian, Moroccan, and Tunisian).

As shown in Table 5.9, both DziriBERT and DarijaBERT perform notably better on Algerian Arabic, achieving accuracies of 98.21% and 94.66%, respectively. In contrast, their performance drops significantly when tested on other dialects, particularly Tunisian and MSA. MDA-BERT , which was pretrained on multi-dialect Arabic, achieves an overall accuracy of 60.29%, lower than both DziriBERT and DarijaBERT. It performs best on Algerian Arabic (88.98%) but struggles significantly on Moroccan and Tunisian.

For Moroccan Arabic, DziriBERT struggled, achieving only 52.60%, while DarijaBERT, pretrained on Moroccan Arabic, performed better, reaching an accuracy of 64.13%. MDA BERT, despite being pretrained on multi-dialect data, achieved only 43.41% on Moroccan Arabic, indicating challenges in generalizing to this dialect. This result was expected, given that DarijaBERT was specifically trained on Moroccan data. However, even DarijaBERT's performance in its native dialect was not high. All three models DziriBERT, DarijaBERT, and MDA BERT struggled even more with Tunisian Arabic, which saw accuracies drop further, with DziriBERT reaching only 65.35%, DarijaBERT 56.04%, and MDA BERT 44.94%. This decrease in performance across dialects highlights the difficulty in generalizing dialect-specific models like DziriBERT and DarijaBERT to other, less similar varieties of Arabic. For MDA BERT, which was pretrained on multi-dialect data, the challenges may stem from the diversity and complexity of the dialects it was exposed to during pretraining, making it harder to achieve high accuracy on specific dialects like Moroccan and Tunisian Arabic with limited data.When it came to MSA, DziriBERT slightly outperformed DarijaBERT and MDA BERT, with accuracies of 70.67%, 66.33%, and 63.72%, respectively. While these results are relatively better than the ones for Tunisian Arabic, they still fall short. The strong performance on Algerian Arabic for both DziriBERT, DarijaBERT and MDA BERT may be partially due to the fact that the Algerian dataset was extracted from a different source compared to the other dialects. The Algerian dataset could possess distinctive features or characteristics that make it easier for the models to classify, potentially explaining the higher accuracies achieved for this dialect. These features might not be as present or recognizable in the datasets for Moroccan, Tunisian, and MSA, leading to the models' poorer generalization to these other dialects.

In contrast, WASL-DI despite being trained on minimal data, achieved impressive results

across all dialects, with an overall accuracy of **92.13%**, significantly higher than DziriBERT, DarijaBERT and MDA BERT, which achieved 71.72%, 70.31% and 60.29%, respectively, WASL-DI shows an exceptional performance in classifying Algerian with 99.90% accuracy. It also achieved good results in Moroccan Arabic at 87.52%, outperforming both DziriBERT, DarijaBERT, and MDA BERT, which, despite their specialized training, achieved lower performance in these dialect. Furthermore, the model achieved 98.19% accuracy in MSA and 82.83% for Tunisian Arabic, indicating strong generalization capabilities. These results proves WASL-DIs ability to generalize from a minimal amount of training data to a much larger and more diverse test set.

5.3.6.2 Arabic Dialects

We also evaluated WASL-DI using various MADAR datasets: MADAR-2 [124], MADAR-6, MADAR-9 [124], and MADAR-26.

- MADAR-2: This dataset combines 25 different dialects into a single "dialect" class, contrasting with MSA. This allows us to assess the models ability to differentiate between a wide range of dialects and MSA.

- MADAR-6: This dataset includes six classes: Doha, Beirut, Cairo, Tunis, Rabat, and MSA.

- MADAR-9: In this dataset, sentences are grouped by region, resulting in nine classes: MSA, Yemen, Maghrebi (including Tunisia, Morocco, and Algeria), Egypt, Libya, Gulf (covering KSA, UAE, Qatar, Bahrain, Oman, and Kuwait), Sudan, Iraq, and Levant (including Syria, Lebanon, Jordan, and Palestine).

- MADAR-26: This dataset features 25 distinct dialects from across the Arab world, in addition to MSA.

Models		[124]			WASL-DI
		ArBERT-based	CAMeL-based	MDA-based	
Madar-2	Accuracy	98.37 (±0.04)	98.21 (±0.14)	98.25 (±0.14)	**99.03 (±0.08)**
	F1-score	87.98 (±0.48)	86.67 (±1.15)	86.67 (±0.35)	**93.54 (±0.55)**
Madar-6	Accuracy	91.09 (±0.17)	91.20 (±0.29)	91.20 (±0.29)	**91.88 (±0.11)**
	F1-score	91.10 (±0.18)	91.21 (±0.30)	91.21 (±0.30)	**91.89 (±0.10)**
Madar-9	Accuracy	79.28 (±0.50)	79.60 (±0.39)	78.60 (±0.39)	**79.95 (±0.18)**
	F1-score	75.75 (±0.64)	75.42 (±0.37)	75.42 (±0.37)	**77.09 (±0.30)**
Madar-26	Accuracy	57.57 (±0.48)	61.66 (±1.00)	58.71 (±0.12)	**62.35 (±0.19)**
	F1-score	57.45 (±0.49)	61.59 (±1.12)	58.77 (±0.09)	**62.33 (±0.15)**

Table 5.10: Performance Comparison of Various Models on the Madar Dataset, Including Results from [124] and Our Proposed Model. The metrics presented are Accuracy and F1-score across five runs with different random seeds, along with the standard deviation across different Madar subsets (Madar-2, Madar-6, Madar-9, and Madar-26).

WASL-DI's consistently outperformed the other models across all MADAR datasets, as shown in Table 5.10. The results underscore the effectiveness and adaptability of our approach in handling the linguistic nuances in diverse dialectal variations.

To further validate the effectiveness of our approach, we conducted a comparative evaluation against the work of Alsuwaylimi [33], which used different architectures and dataset. Their dataset [32] included four distinct dialects: Egyptian, Tunisian, Yemeni, and Jordanian, with approximately 38k samples. The models combined CAMeLBERT and ALBERT with BiLSTM, focusing on various language tasks.

This comparison is particularly interesting because it highlights how different architectural choices can influence the performance of dialect identification models. While both our and their studies employed transformer-based model (CAMeLBERT) with a traditional deep learning approach BiLSTM, our approach incorporates CAMeLBERT with BiGRU, which processes outputs in parallel and concatenates them, in contrast to their sequential approach of passing CAMeLBERT's output into a BiLSTM. This architectural difference likely contributed to the superior performance of WASL-DI, as detailed in Table 5.11.

These results emphasize the superior performance and generalizability of our hybrid architecture.

5.4 Error Analysis and Model Limitations

Our model was designed to distinguish between MSA and the Algerian dialect, aiming for robust performance across different language scripts. To understand the model's classification behavior,

Models	[33]		WASL-DI
	CAMeLBERT + BiLSTM	ALBERT + BiLSTM	
Accuracy	87.67	86.51	**88.41 (±0.22)**
F1-score	87.76	86.69	**88.49 (±0.17)**

Table 5.11: Performance Comparison of Different Models on the Task, Highlighting Results from [33] Using CAMeLBERT and ALBERT with BiLSTM Architectures, Alongside Our Proposed Model. Metrics include Accuracy and F1-score.

we conducted an error analysis, beginning with an examination of the dataset composition.

The dataset includes sentences in two distinct scripts: Arabic and Arabizi, with Arabizi only present in the Algerian dialect. This distribution raised questions about the influence of script on classification, as visualizing the data showed a notable imbalance, with most sentences written in Arabic. This script usage pattern may have influenced the models behavior, leading it to associate Arabizi almost exclusively with the Algerian dialecta key point that emerged as we analyzed the misclassifications.

Figure 5.6 presents the distribution of arabizi and arabic text across both categories MSA and algerian.

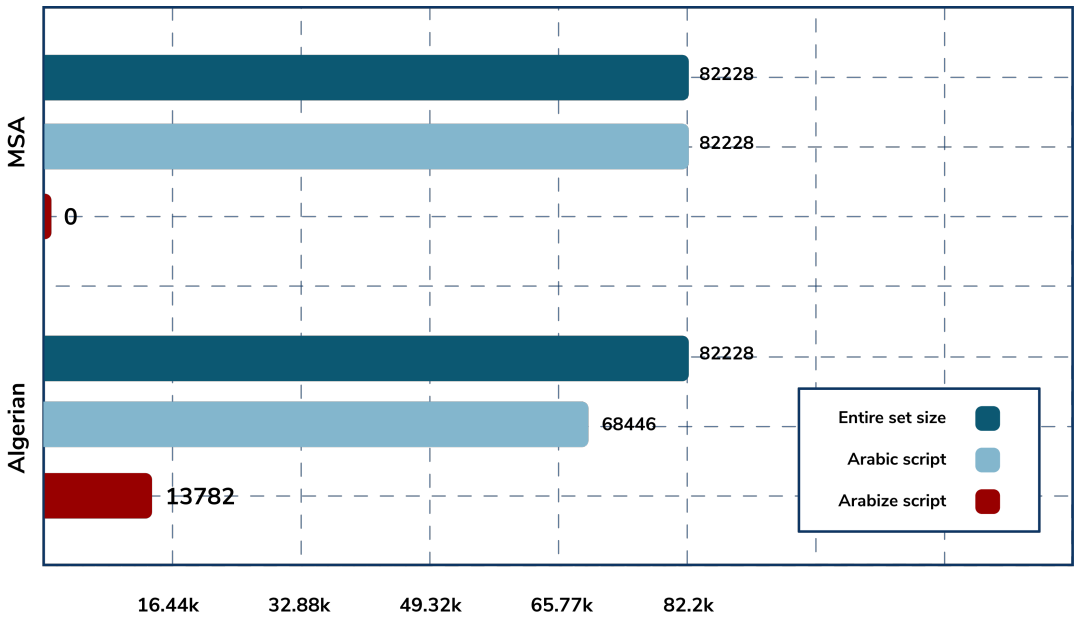

Figure 5.6: Distribution of Arabizi and Arabic text across MSA and Algerian categories.

As highlighted in figure 5.6, the overwhelming majority of the sentences were written in Arabic, with Arabizi representing only a small proportion of the Algerian dialect instances. This skewed distribution raises questions about its impact on the model's behavior, particularly in how it may influence the association of Arabizi with Algerian.

The model's confusion matrix presented in Figure 5.7 demonstrates its classification performance for MSA and Algerian dialect sentences, considering both Arabic script and Arabizi script.

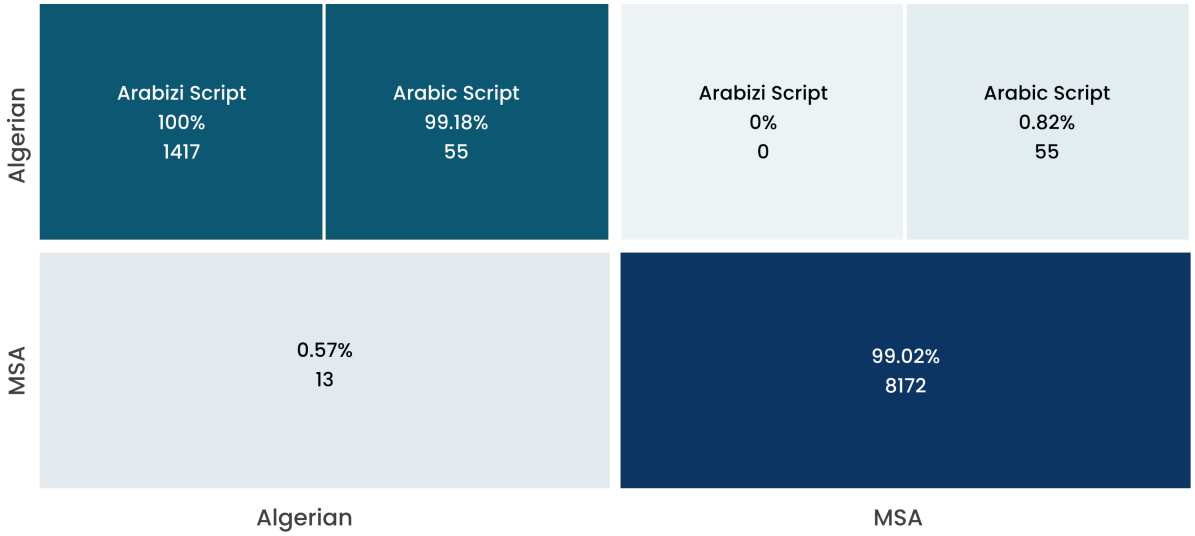

Figure 5.7: Confusion matrix of the model's classification performance for MSA and Algerian dialect sentences based on script (Arabic script vs. Arabizi script). The rows represent the true labels, while the columns indicate the predicted labels.

The confusion matrix in 5.7 reveals that the model heavily relies on script as a key indicator of dialect, achieving perfect accuracy (100%) for Algerian sentences written in Arabizi script, with 1,417 instances and no misclassifications in this category. This suggests that Arabizi script may serve as a strong marker for identifying the Algerian dialect, as all misclassified instances occurred among sentences written in Arabic script.

To assess the model's sensitivity to Arabizi, we conducted an experiment where a selection of MSA sentences, originally written in Arabic, were chosen from the test set, specifically those that the model had already classified correctly. These sentences were then transliterated into Arabizi using Buckwalter transliteration [59], and input back into the model to observe any shift in classification behavior.

This setup aimed to reveal whether the model relied heavily on script for its predictions or if it could generalize based on linguistic features that are independent of the writing system.

The results of this experiment are summarized in table 5.12.

In the original form, all sentences were accurately classified as MSA, reflecting the models ability to identify them when written in Arabic script. However, upon transliteration to Arabizi, each of these sentences was misclassified as belonging to the Algerian dialect, despite the linguistic content remaining unchanged. This consistent misclassification across the sentences suggests a clear dependency on script as a distinguishing feature. The model appears to associate Arabizi exclusively with the Algerian dialect, irrespective of actual linguistic indicators within

Original Sentence	Ground truth	Prediction	Transliteration	Transliteration Prediction
عن بلد أصابها العفن والفساد والجهل	MSA	MSA	Eano baladK >aSabaha AlEafanu waAlfasaAdu waAlojaholu	Algerian
القصة ممتعة جدا جدا جدا ومفيدة حتى على المستوى الثقافي لإنها بتعتمد على معلومات كثيرة حقيقية	MSA	MSA	AloqiS apu mumotiEapN jid~FA jid~FA jid~FA wamufiydapN Hat~aY EalaY AlomusotawaY AlovaqaAfiy~i li>n~ahaA botaEotamidu EalaY maEoluwmaAtK kaviyrapK Haqiyqiy~apK	Algerian
جميل ولكني أحببت الجزء الأول أكثر	MSA	MSA	jamiylN walakin~iy >aHobabotu Alojuzo'a Al>w~ala >akovara	Algerian
لغة منيف فاتنة، وسرده سلس ممتع. لا شك أني أسفت	MSA	MSA	lugapu maniyfK fatinapN wasaroduhu salisN mumotiEN. laA $ak a >an~iy >asifotu	Algerian

Table 5.12: Examples of Original Sentences with Ground Truth and Model Predictions Before and After Transliteration. This table compares the classification outcomes for Arabic script and Arabizi transliterations of both MSA.

the sentence. Rather than focusing on vocabulary, syntax, or morphological differences that are independent of script, the model has developed a bias towards identifying Arabizi as an Algerian marker.

To further examine the model's reliance on script, and whether the use of Arabic script affects the model's performance on Algerian dialect, we conducted a complementary experiment by transliterating a selection of Algerian sentences originally written in Arabizi into Arabic script. We chose sentences from the test set that the model had previously classified correctly as Algerian when they were in Arabizi. After transliterating these sentences to Arabic script, we re-evaluated them with the model to observe if the change in script would lead to any misclassification.

The results are presented in table 5.13.

In contrast to our previous experiment, the predictions presented in Table 5.13 show that the model correctly classified the Arabic-script versions of these Algerian sentences as Algerian dialect. This indicates that, despite the model's tendency to associate Arabizi with Algerian, it can still recognize Algerian dialectal features when they are presented in Arabic script. This suggest that, while the model does have a strong association between Arabizi and the Algerian dialect, it is still able to capture some inherent linguistic characteristics of Algerian that are independent of script, which demonstrates a level of generalization within the model that allows

Original Sentence	Ground truth	Prediction	Transliteration	Transliteration Prediction
Yakhi chiyat	Algerian	Algerian	ياخي شيات	Algerian
diralna 208 wal clio 4	Algerian	Algerian	ديرلنا 802 ولا كليو 4	Algerian
bral andah golf 7 w yadra maya3rafch yahdar	Algerian	Algerian	بغل عنده غولف 7 و يادرا ميعرفش يهدر	Algerian
vrm mrç rabi y3tik ma tatmana	Algerian	Algerian	فريمون مارسي ربي يعطيك ما تمتني	Algerian

Table 5.13: Examples of Original Sentences with Ground Truth and Model Predictions Before and After Transliteration. This table compares the classification outcomes for Arabic script and Arabizi transliterations of Algerian dialect sentences.

it to maintain accuracy on Algerian text even when the script changes to Arabic.

The model's inconsistent reliance on script-specific cues raises practical concerns, especially in real-world applications where both MSA and Arabic dialects can appear in either Arabic or Arabizi script. This script-switching is common on social media platforms, where users frequently shift between scripts, making it crucial for models to generalize effectively across both formats.

5.4.1 Isolating Script Influence: A Balanced Dataset Experiment

To verify whether the model's strong association between Arabizi and the Algerian dialect was due to an inherent limitation or simply a result of dataset imbalance, we extended our investigation by constructing a more controlled setup. As shown in Figure 5.6, Arabizi was overwhelmingly associated with the Algerian class—only about 13K out of 82K Algerian sentences were written in Arabizi, while MSA contained none. To mitigate this imbalance and test the assumption that the issue was data-driven, we transliterated an additional 30K Algerian sentences from Arabic script into Arabizi, increasing Arabizi representation within the Algerian class to approximately half of the Algerian data. We also transliterated 40K MSA sentences into Arabizi to introduce this script into the MSA class for the first time. This adjustment resulted in a balanced dataset where both scripts (Arabic and Arabizi) appeared in both classes (MSA and Algerian) almost equally, allowing us to isolate script influence and determine whether the model's behavior was shaped by script distribution in the data or by deeper model limitations.

After training the model on this balanced dataset, we observed a significant shift in classification behavior. The model no longer misclassified MSA sentences as Algerian when written in Arabizi, nor did it rely solely on script as a distinguishing feature. In fact, it correctly classified all the instances—from the previous experiment 5.12 and 5.13—in both their original and transliterated forms. This demonstrates that the model is now relying on linguistic content rather than script alone, providing strong evidence that the initial misclassifications were the result of

dataset imbalance rather than an inherent model limitation.

However, we observed a slight drop in overall accuracy compared to the original dataset. As shown in Table 5.14, the models accuracy decreased to 98.27%, a minor reduction in exchange for eliminating script-based bias.

Table 5.14: Performance of WASL-DI on balanced data.

Data	Accuracy%	MSA Accuracy%	Algerian Accuracy%
Original Dataset	99.24	99.01	99.46
Balanced Dataset	98.27	98.18	98.35

This experiment provides strong evidence that the original models reliance on script was a dataset-driven problem rather than a fundamental flaw in its design. By ensuring that both scripts (Arabic and Arabizi) appeared in both categories (MSA and Algerian) almost equally, we enabled the model to generalize better across scripts. Although the model's accuracy slightly decreased, this trade-off eliminated script-based bias, making the model more robust.

5.5 Ablation Study

To comprehensively evaluate the significance of each component within WASL-DI architecture, we conducted a thorough ablation study to understand how individual elements contribute to overall performance, allowing us to isolate the effects of specific features. We began by assessing the impact of using pre-trained weights from BERT and FastText, the goal of this phase was to determine whether the inclusion of these pre-trained weights significantly enhances the model's ability to accurately identify dialects compared to using randomly initialized weights.

Following this, we performed a component-wise ablation analysis to investigate the contributions of specific model components. By incrementally adding components and comparing the model's performance against both the full architecture (to assess the impact of the added component) and the previous model[1] (to gauge progress towards the goal), we aimed to identify which parts were critical for achieving optimal accuracy. Each configuration was trained multiple times using different random seeds to ensure robust results.

We trained each model five times with different random seeds to mitigate the effects of random initialization. To evaluate statistical significance, we computed McNemar's test for each seed and paired t-tests across the five runs. We employed these tests to assess the significance of performance differences, providing a comprehensive understanding of how individual features and pre-trained weights influenced dialect identification.

[1]The term 'Previous Model' refers to the model mentioned directly above in the table, which serves as the baseline for comparison in the subsequent row.

5.5.1 Impact of Pre-trained BERT and FastText Weights

Table 5.15 presents the outcomes of training WASL-DI using random initialization for both the BERT model and the Embedding Layer.

The results indicate a significant performance drop when compared to the full model that uses pre-trained weights. Specifically, the accuracy values drop considerably, and the associated p-values from McNemar's test from each seed and the one from paired t-test strongly indicate that the differences to the full architecture are statistically significant.

Seed	Accuracy	McNemar's p-value	Paired t-Test p-value
42	97.74	3.46×10^{-41}	0.3×10^{-3}
200	97.77	5.23×10^{-43}	
250	98.04	8.49×10^{-37}	
300	98.11	5.85×10^{-35}	
350	98.29	1.50×10^{-24}	

Table 5.15: Impact of Random Initialization on Model Performance with BERT and FastText. This table displays the accuracy results obtained from training the model with randomly initialized BERT weights and the corresponding p-values from McNemar's test and paired t-test.

5.5.2 Component-wise Ablation

In our component-wise ablation study, we systematically evaluated different configurations of WASL-DI to understand the contributions of each architectural element. Starting from a basic CNN architecture and progressively integrating additional layers, to assess their impact on performance.

The results demonstrate that the CNN-only model exhibits a significant performance drop, highlighting its limitations in capturing the complexities of dialect identification. By integrating a GRU layer, we observed a marked improvement in accuracy, supported by statistically significant p-values from both McNemar's test and paired t-tests.

Further enhancements come from incorporating an attention mechanism, which allows the model to focus on relevant parts of the sequence. This leads to even higher accuracy.

Ultimately, the full model, which incorporates CAMeL BERT, achieved the highest accuracy, confirming the effectiveness of leveraging advanced embeddings alongside well-designed architectural components.

5.6 Conclusion

In this part of our research, we proposed a hybrid model that leverages cross-lingual transfer learning to address the challenges of dialect identification between MSA and Algerian Arabic,

Ablation Model	Seed	Accuracy	McNemar's p-value (vs. Full Model)	McNemar's p-value (vs. Previous Model)	Paired t-Test p-value (vs. Full Model)	Paired t-Test p-value (vs. Previous Model)
CNN	42	94.56	4.02×10^{-186}	—	8.66×10^{-7}	
	200	94.04	1.72×10^{-214}	—		
	250	94.48	4.3×10^{-190}	—		
	300	94.56	2.18×10^{-193}	—		
	350	94.28	5.73×10^{-201}	—		
Previous Model + MLP	42	96.36	1.22×10^{-100}	1.08×10^{-26}	1.67×10^{-6}	9.65×10^{-5}
	200	96.27	3.16×10^{-107}	1.21×10^{-51}		
	250	96.04	2.12×10^{-118}	7.24×10^{-17}		
	300	96.29	6.6×10^{-112}	2.27×10^{-39}		
	350	96.34	1.09×10^{-102}	5.89×10^{-35}		
Previous Model + GRU	42	96.71	4.26×10^{-80}	0.3×10^{-2}	3.11×10^{-5}	0.3×10^{-2}
	200	97.20	2.86×10^{-63}	1.06×10^{-16}		
	250	97.21	1.23×10^{-66}	3.6×10^{-24}		
	300	97.28	1.08×10^{-62}	5.29×10^{-20}		
	350	97.18	9.38×10^{-68}	1.97×10^{-15}		
Previous Model + Attention	42	97.43	6.56×10^{-53}	1.48×10^{-11}	5.93×10^{-8}	0.15×10^{-1}
	200	97.49	7.3×10^{-50}	0.7×10^{-3}		
	250	97.49	3.68×10^{-41}	0.4×10^{-3}		
	300	97.50	4.29×10^{-49}	0.13×10^{-1}		
	350	97.50	2.75×10^{-52}	0.19×10^{-3}		
Full Model (Previous Model + CAMeL BERT)	42	99.23	—	—	—	—
	200	99.22	—	—		
	250	99.28	—	—		
	300	99.24	—	—		
	350	99.21	—	—		

Table 5.16: Results of the component-wise ablation study, showcasing the accuracy and statistical significance of various model configurations. The values reported are the accuracy across five runs with different random seeds, along with the tests $p - values$

a low-resource dialect. By utilizing the extensive linguistic resources available in MSA, our approach compensates for the scarcity of annotated data, a common limitation in dialectal NLP. Our model combines pre-trained language representations with deep learning techniques, capturing both contextual and dialect-specific features. As a result, the model demonstrates superior performance compared to baseline models, and previous approaches. Its ability to generalize across diverse dialects and maintain stability with limited training data showcases its adaptability and effectiveness. Through this chapter, we directly addressed Research Questions 2, 3, and 4, by evaluating the generalizability of MSA-based models to dialectal Arabic, examining

their ability to distinguish dialectal features, and testing the benefit of combining complementary representations to enhance dialect modeling.

Our work contributes substantial progression NLP for low-resource languages, highlighting the potential of cross-lingual transfer learning from high-resource languages. By adapting MSA resources to dialects with limited data, our approach addresses both the scarcity of annotated data and the linguistic diversity across dialects. The integration of MSA embeddings with specialized model layers results in a scalable architecture capable of capturing both high-level language patterns and the unique nuances specific to dialects. This approach not only narrows the gap between high- and low-resource languages but also offers a flexible framework applicable to various challenges in both dialectal and low-resource NLP tasks. Furthermore, the adaptability of this framework has broad implications for industries that rely on NLP, such as automated customer support, sentiment analysis, and content moderation, particularly in dialect-rich regions. Ultimately, this work extends NLP capabilities to under-resourced dialects and languages, promoting greater accessibility, inclusivity, and empowerment for speakers of underrepresented dialects and languages.

In conclusion, this research demonstrates the power of cross-lingual transfer learning in overcoming the challenges of low-resource dialect identification. By harnessing MSA's linguistic resources, we have developed a robust and adaptable model that addresses the limitations faced by dialect-rich regions, especially those with scarce annotated data. This approach not only contributes to the field of NLP but also lays the groundwork for future research aimed at creating more inclusive language technologies for diverse global communities.

While this chapter relies on pre-trained MSA models to transfer linguistic knowledge, the next chapter advances this approach by directly harnessing MSA data through a multitask and multisource learning framework. This framework simultaneously performs sentiment analysis on Algerian Arabic while utilizing MLM on MSA, allowing the model to dynamically share linguistic features between the two tasks. By integrating a shared MoE layer with self-attention mechanisms, this approach aims to capture both dialect-specific nuances and general linguistic patterns from MSA, further enhancing performance across diverse dialectal and task settings.

CHAPTER 6

Multitask Learning for Sentiment Analysis in Algerian Arabic: A Transfer-Based Framework SILAA-SA

This chapter introduces SILAA-SA, a multitask learning framework designed to address sentiment analysis in Algerian Arabic. Building on the foundations of cross-lingual transfer and hybrid modeling, SILAA-SA integrates shared knowledge between MSA and dialectal data through a Mixture-of-Experts architecture. The chapter includes extensive evaluations across several datasets and tasks to validate the model's generalizability and scalability.

Building on prior work that leveraged pretrained linguistic knowledge from MSA to improve dialect identification, this chapter adopts a distinct approach by focusing on utilizing data resources. While the previous chapter demonstrated the effectiveness of hybrid models combining CAMeLBERT and FastText embeddings to transfer knowledge across linguistic varieties, the current exploration shifts the focus toward harnessing diverse datasets to address dialectal variability.

Data-driven methodologies provide a complementary avenue for enhancing dialect identification, particularly when large-scale annotated resources are available. Unlike approaches dependent on pretrained models, which rely on prior linguistic representations, data-centric strategies leverage extensive and diverse samples to capture dialect-specific nuances directly. This paradigm acknowledges that while pretrained models are valuable for transferring generalized knowledge, a robust collection of dialectal data can expose fine-grained variations that are otherwise difficult to capture.

To address these limitations, we propose SILAA-SA (Shared Integration of Low-resource Algerian and Arabic for Sentiment Analysis), a novel multitask learning framework designed to enhance sentiment analysis in low-resource dialects by leveraging MSA data as an auxiliary task. The name "SILAA" is derived from the Arabic word "sila" (صِلَةٌ), meaning "connection," reflecting

its role in linking MSA and Algerian Arabic. Unlike traditional transfer learning, which relies solely on fine-tuning for a single task, SILAA-SA introduces a multitask learning paradigm that simultaneously performs sentiment analysis on Algerian Arabic and MLM on MSA. This approach allows the model to leverage the rich linguistic resources of MSA while adapting to the unique characteristics of Algerian Arabic.

SILAA-SA integrates QARiB BERT embeddings [3] and self-attention mechanisms to process both Algerian Arabic and MSA inputs. The model uses separate BERT encoders for sentiment analysis (Algerian Arabic) and masked language modeling (MSA), each followed by a self-attention layer to capture contextual relationships. The outputs from both tasks are then fed into a shared MoE layer [95, 98], which dynamically combines the knowledge from sentiment analysis and MLM using a gating mechanism. Finally, the outputs from the MoE layer are passed through task-specific heads: a classification layer for sentiment analysis and an MLM head for masked language modeling. This architecture allows SILAA-SA to effectively capture shared linguistic features while adapting to the unique characteristics of Algerian Arabic, offering a promising solution to the challenges of low-resource NLP.

The key innovation of SILAA-SA lies in its ability to bridge the gap between high-resource MSA and low-resource Algerian Arabic. By employing a multitask learning paradigm, The shared MoE layer dynamically balances shared linguistic features and dialect-specific nuances, enabling the model to generalize effectively across datasets and tasks.

This analysis builds on insights from the hybrid cross-lingual model in the previous chapter while presenting a comparative framework that emphasizes data as a primary driver of model performance. Through systematic experimentation, we evaluate the extent to which richer and more representative datasets contribute to improving classification accuracy and cross-dialect generalization, offering a complementary perspective on advancing Arabic dialect NLP beyond the confines of pretrained model dependence.

Our contributions in this chapter are as follows:

- We introduce SILAA-SA, a multitask learning framework designed to enhance sentiment analysis in low-resource Algerian Arabic by leveraging the already available resources of MSA. Unlike traditional transfer learning approaches, which rely solely on limited dialect-specific data, SILAA-SA uses MSA as an auxiliary task to enrich the model's linguistic knowledge. By fine-tuning on QARiB BERT (a model trained on both MSA and dialectal data) and leveraging MSA's rich syntactic and semantic resources, SILAA-SA significantly improves performance and generalization for low-resource dialects.

- We propose a novel architecture that integrates BERT embeddings, self-attention mechanisms, and a shared MoE layer. This architecture dynamically balances shared linguistic features and dialect-specific nuances, improving generalization across datasets and tasks.

- We conduct extensive experiments on multiple datasets, including Algerian, Tunisian, and Moroccan Arabic, demonstrating Silaa-SA's cross-dialect generalization and superior performance in sentiment analysis. Additionally, we evaluate Silaa-SA on fake news detection, showcasing its cross-task generalization and ability to adapt to different NLP tasks. This dual capability highlights Silaa-SA's robustness and versatility in low-resource settings.

- We provide a comprehensive analysis of the model's performance, including ablation studies, to identify the contributions of individual components.

Publication Note

The work presented in this chapter has been submitted as a journal paper and is currently under review in **IEEE Transactions on Affective Computing**.

- M. Chabane, F. Harrag, and K. Shaalan. (2025). SILAA-SA: A Multitask Mixture-of-Experts Framework for Sentiment Analysis in Low-Resource Arabic Dialects via Modern Standard Arabic Transfer. IEEE Transactions on Affective Computing. Under review.

6.1 Related Works

Sentiment analysis has been the focus of various studies, as outlined in the related work section (see Chapter 3, particularly Section 3.2).

In addition to the studies previously mentioned, we incorporated insights from the following three studies that extend the application of sentiment analysis in Arabic dialects, which have provided new methodologies and findings that further refine sentiment classification techniques, particularly for underrepresented dialects.

For instance Abdedaiem et al. [2] introduced FASSILA, a specialized corpus designed for Fake News detection and Sentiment Analysis in the Algerian Dialect. The corpus comprises 10,087 sentences with over 19,497 unique words. FASSILA covers seven domains, including politics, health, and sports. The authors conducted experiments using BERT-based models (e.g., AraBERT, DziriBERT, MarBERT) and traditional Machine Learning models (e.g., SVM, Logistic Regression, Decision Trees). Among the transformer-based models, AraBERTv02 performed best for sentiment analysis with an accuracy of 83.92% and an F1-score of 80.35%. For machine leanring models, SVM showed consistent performance across embeddings, particularly with AraBERTv02, achieving the best performance across all models with an accuracy of 84.42% and an F1-score of 80.96% for sentiment analysis.

Rahab et al. [133] conducted sentiment analysis on comments from Algerian newspaper websites, focusing on classifying opinions into positive, negative, and neutral categories. The study utilized two corpora: SANA, a newly created corpus of comments from three Algerian

newspapers, and OCA, a publicly available Arabic sentiment analysis corpus. The authors employed three classifiers—SVM, Naïve Bayes, and KNN—and evaluated their performance using 10-fold cross-validation. Naïve Bayes achieved the highest accuracy on both corpora: 89.80% on the OCA corpus and 75.00% on the SANA corpus.

the authors in [118] experimented with sentiment analysis in the context of Tunisian Arabic, a dialect that poses unique challenges. They emphasize the role of deep learning architectures like LSTM and BERT-based models, in improving sentiment classification accuracy. The study notes that traditional sentiment analysis tools designed for MSA often underperform on Tunisian dialect data due to its distinct morphological and lexical characteristics. The authors propose leveraging transfer learning and dialect-specific embeddings to enhance model performance, using TSAC dataset [117] their best model is TunoBert-CNN which achieved an accuracy of 90.8% and an F1-score of 91.05%.

Work	**Data**	**Model**	**Class**	**Accuracy %**	**F1-score %**
Abdedaiem et al. [2]	Fassila	AraBERTv02	3	83.92	80.35
		SVM with AraBERTv02 Embedding		**84.42**	**80.96**
Rahab et al. [133]	SANA	Naïve Bayes	3	70.00	—
	SANA Without Neutral		2	75.00	—
	OCA		2	89.80	—
Mechti et al. [118]	TSAC	TunoBERT-CNN	2	90.80	91.05

Table 6.1: Summary of Additional Sentiment Analysis related works.

6.2 Methodology

In this section, we present the methodologies and models we used in the sentiment analysis. We begin by describing the baseline model architectures, followed by the introduction of our novel multitask learning framework. A detailed explanation of the architectures is provided, along with the hyperparameters and training procedures adopted to ensure consistent and robust evaluation across all experiments.

6.2.1 Hyperparameters Summary

To ensure consistency and fairness, all models were trained using the same set of hyperparameters. Table 6.2 summarizes the key hyperparameters used for both the baseline models, and the proposed multi-task approach.

Parameters	Value
Batch size	32
Learning rate	8×10^{-6}
Optimizer	AdamW
Sequence Length	50

Table 6.2: The hyperparameters of our models

6.2.2 Baselines

In this section, we present the baseline architectures for our study. Three variants of the architecture are proposed, each using a different layer: BiGRU, BiLSTM, or CNN. All architectures leverage QARiB BERT embeddings as the foundation for capturing contextualized representations of the input text. Additionally, we explore a variant of each baseline that incorporates a self-attention mechanism.

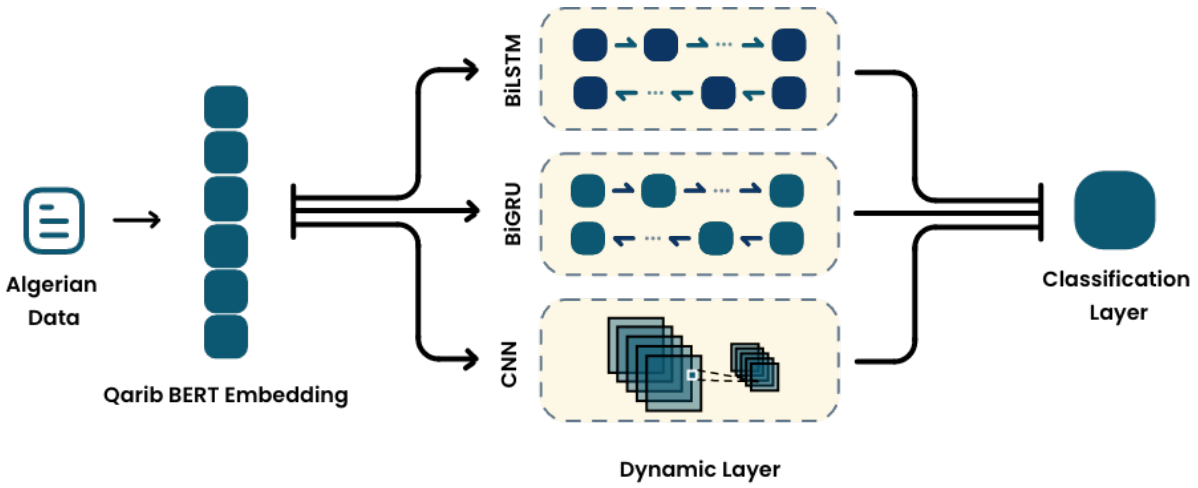

Figure 6.1: Baseline unitask learning architecture with self-attention. The architecture combines QARiB BERT embeddings, and a dynamic layer (BiGRU, BiLSTM, or CNN).

The architecture without self-attention, shown in Figure 6.1, begins with QARiB BERT embeddings, which provide rich contextualized representations of the input text. These embeddings are passed through a dynamic layer, implemented as one of three options:

- **BiGRU:** A bidirectional Gated Recurrent Unit (GRU) that processes the sequence in both forward and backward directions to capture temporal dependencies.

- **BiLSTM:** A bidirectional Long Short-Term Memory (LSTM) network that enhances the model's ability to capture long-range dependencies in the input sequence.

- **CNN:** A Convolutional Neural Network that extracts local features from the sequence using convolutional filters.

The output of the dynamic layer is passed through a fully connected layer for classification.

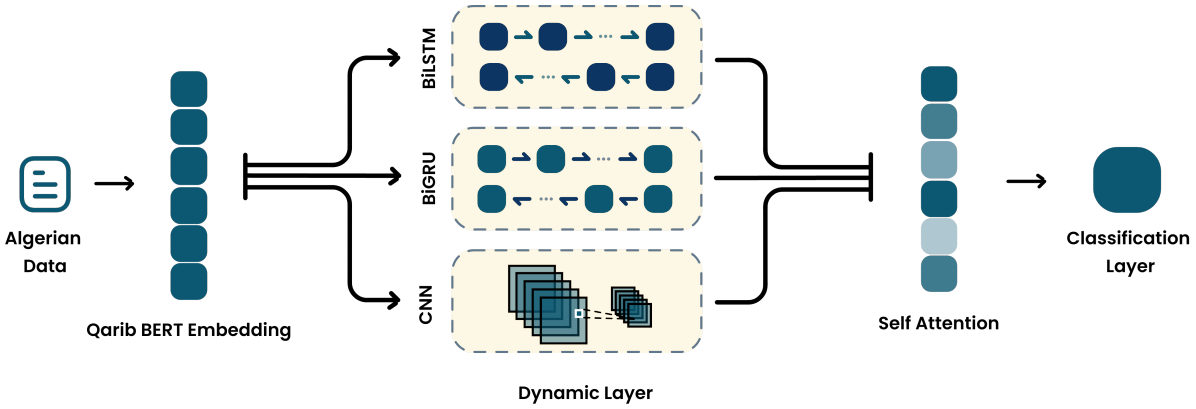

Figure 6.2: Baseline unitask learning architecture with self-attention. The architecture combines QARiB BERT embeddings, a dynamic layer (BiGRU, BiLSTM, or CNN), and a self-attention layer.

The architecture with self-attention, depicted in Figure 6.2, extends the previous model by incorporating a self-attention layer. Similar to the first architecture, it begins with QARiB BERT embeddings, which are processed by a dynamic layer (BiGRU, BiLSTM, or CNN). The output of the dynamic layer is then passed through a self-attention layer, which computes attention scores to weigh the importance of different parts of the input sequence. This allows the model to focus on the most relevant features for the task. The attended output is fed into a fully connected layer for final classification.

6.2.3 Proposed Approach

In this section, we describe the proposed multitask learning model, designed primarily for sentiment analysis. To enhance the model's ability to capture linguistic nuances, we incorporate MLM as a helper task. The MLM task facilitates knowledge sharing and improves the model's understanding of contextual features, though its predictions are not the primary focus. Figure 6.3 illustrates the architecture of our proposed model.

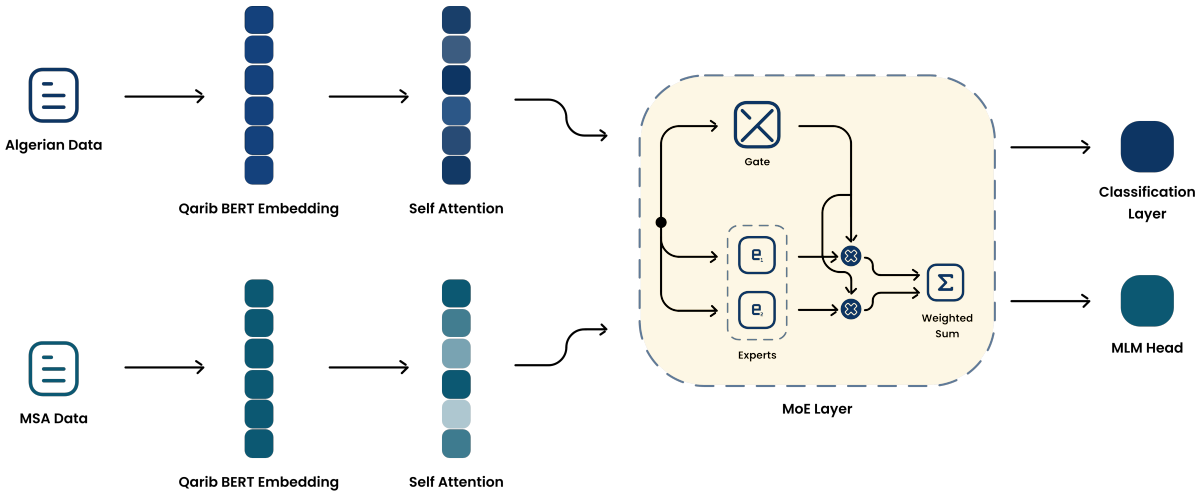

Figure 6.3: Proposed multitask learning architecture combining BERT embeddings, self-attention, and a shared MoE layer.

The architecture integrates Qarib-BERT embeddings, task-specific self-attention layers, and a shared MoE layer to effectively capture both global contextual and task-specific features. Below, we describe the key components and their roles in the architecture.

BERT Encoders

- Sentiment Analysis (Main Task): A Qarib-BERT model processes the input sequence to produce contextualized embeddings. These embeddings capture the semantic and syntactic information of the input text.

- Masked Language Modeling (Helper Task): A Qarib-BERT encoder, configured for masked language modeling, processes the input sequence with masked tokens and produces hidden states. While the MLM task is not the primary focus, it serves as a helper task to improve the model's understanding of contextual features, which indirectly benefits the main sentiment analysis task.

Task-Specific Self-Attention Layers

- Each task is equipped with a self-attention layer that refines the token representations by computing attention scores. These scores determine the importance of each token relative to others in the sequence, allowing the model to focus on task-relevant information.

Shared Mixture of Experts Layer

The MoE layer is shared across both tasks and consists of multiple experts, each implemented as a feedforward neural network. A gating mechanism computes weights (gate scores) that determine the contribution of each expert to the final output. These weights are used to combine

the outputs of all experts using a weighted sum, allowing the model to dynamically adjust the importance of each expert based on the input.

- Experts: Each expert processes the input independently, capturing different aspects of the data. This allows the model to learn diverse representations that are useful for both tasks.

- Gating Mechanism: Computes gate scores to determine the contribution of each expert, ensuring that the most relevant experts are prioritized for each input.

- Output: A weighted combination of the expert outputs, which is then passed to the task-specific heads.

Task-Specific Heads

- **Sentiment Analysis**
  - The output of the MoE layer is pooled using global average pooling to reduce the sequence dimension.
  - A dropout layer is applied for regularization.
  - A fully connected layer maps the pooled embeddings to the number of sentiment classes, producing the final sentiment classification output.

- **Masked Language Modeling**
  - The output of the MoE layer is passed through the MLM classification head, which produces logits for predicting the masked tokens.

6.3 Experimentation and Results

This section outlines our experimental framework, covering the datasets utilized for training, the obtained results and their analysis, the validation procedures, an investigation into the generalizability of SILAA-SA.

6.3.1 Dataset

To support robust model training and facilitate effective knowledge transfer from MSA, we complement the Algerian dialectal dataset with a well-established MSA corpus. This inclusion enables the model to leverage general Arabic linguistic structures and vocabulary, bridging the gap between resource-rich MSA and the under-resourced Algerian dialect.

6.3.1.1 Algerian Dataset

We used the FASSILA Corpus [2], a specialized dataset for Sentiment Analysis and Fake News detection, in the Algerian Dialect. The corpus consists of 10,087 sentences with over 19,497 unique words, covering seven domains: politics, health, sports, car prices, tourism, eCommerce, and car accidents. Data was collected from diverse sources, including social media, existing datasets, handcrafted sentences by native Algerian speakers, and GPT-4-generated paraphrased sentences.

For Fake News detection, the corpus contains 5,393 real and 4,694 fake sentences, ensuring a balanced distribution. Although Fake News detection is not the main focus of this study, we later utilize this portion of the dataset to assess the generalizability of our model across different tasks. For Sentiment Analysis, sentences are labeled as positive, negative, or neutral. Figure 6.4 illustrates the distribution of sentiment labels across the dataset, providing a clear overview of the data composition.

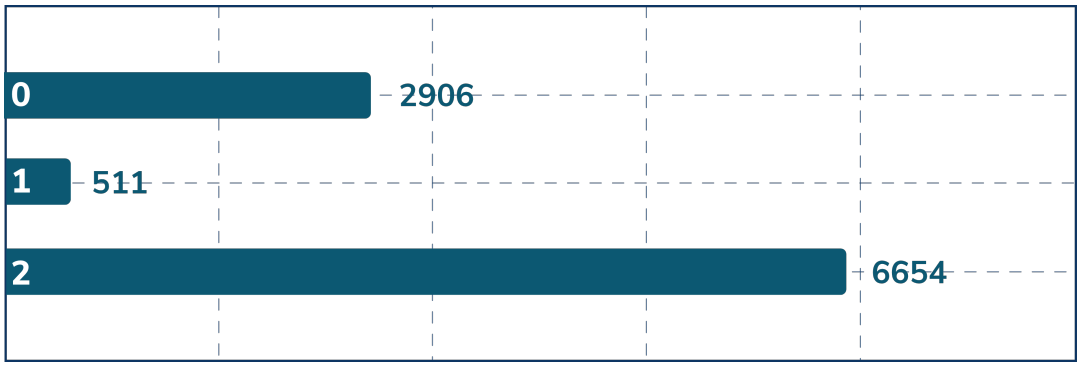

Figure 6.4: The distribution of sentiment labels across the dataset.

6.3.1.2 MSA Dataset

In addition to the FASSILA Corpus, we utilized the HARD (Hotel Reviews Arabic Dataset) [77] for the MLM task. The HARD dataset consists of hotel reviews in Arabic. The dataset includes a wide range of reviews, providing rich and varied language patterns.

To ensure a balanced training phase, we incorporated an equivalent number of samples from the HARD for the MLM task.

6.3.2 Results and Discussion

In this section, we present and discuss the experimental results obtained from evaluating various baseline models and ou proposed model, on the task of sentiment analysis. The performance of these models is assessed using standard evaluation metrics, including accuracy, F1-score, recall, and precision.

6.3.2.1 Baseline Results

Model	Acc. %	F1-Score %	Recall %	Precision %
BERT-BiGRU	84.57	78.53	79.01	79.36
BERT-BiGRU with attention	85.35	**80.31**	80.48	**81.77**
BERT-BiLSTM	83.59	77.67	79.50	78.32
BERT-BiLSTM with attention	83.40	78.47	80.19	78.75
BERT-CNN	**85.55**	78.81	79.76	80.33
BERT-CNN with attention	84.28	78.69	**81.48**	78.27

Table 6.3: Performance comparison of different baseline models architectures on the task, evaluated using accuracy Accuracy, F1-score, recall, and precision. The models include BERT combined with BiGRU, BiLSTM, and CNN, both with and without attention mechanisms. The best-performing values for each metric are highlighted in bold.

The results presented in Table 6.3 highlight the performance of our baselines, BERT-based architectures in conjunction with BiGRU, BiLSTM, and CNN, evaluated using accuracy, F1-Score, recall, and precision. Among the architectures, BERT-BiGRU with attention demonstrates superior performance in terms of F1-Score (80.31%) and precision (81.77%), indicating that the attention mechanism effectively enhances the GRU's ability to balance precision and recall. The BERT-CNN model achieves the highest accuracy (85.55%). Interestingly, when paired with attention, the CNN model achieves the highest recall (81.48%), suggesting improved sensitivity in identifying sentiments but at the cost of a slight reduction in precision and accuracy.

The introduction of attention mechanisms consistently benefits BiGRU and BiLSTM models, as seen by improvements in precision and recall metrics. However, the effect of attention on CNN-based models is more nuanced, with noticeable gains in recall but slight trade-offs in other metrics.

Overall, the results illustrate that attention mechanisms, when carefully integrated, can significantly enhance performance, particularly in recurrent architectures like BiGRU and BiLSTM. The observed variations across metrics further emphasize the importance of aligning model selection with the specific demands of the task.

6.3.2.2 Proposed Approach Results

Table 6.4 presents SILAA-SA's performance on the FASSILA dataset for sentiment analysis. The model achieves an accuracy of 86.81%, along with an F1-Score of 84.46%, a recall of 83.74%, and a precision of 85.44%.

Model	Acc. %	F1-Score %	Recall %	Precision %
SILAA-SA	**86.81**	**84.46**	**83.74**	**85.44**

Table 6.4: Performance of SILAA-SA on the FASSILA dataset for sentiment analysis. Metrics include Accuracy, F1-Score, Recall, and Precision.

Compared to the best-performing baseline models, SILAA-SA achieves notable improvements across all evaluation metrics. While the BERT-CNN model yields the highest accuracy among baselines at 85.55%, SILAA-SA surpasses this with an accuracy of 86.81%. In terms of F1-score, SILAA-SA attains 84.46%, outperforming the BERT-BiGRU with attention, which had the best F1-score among the baselines at 80.31%. Similarly, SILAA-SA achieves higher recall (83.74% vs. 81.48%) and precision (85.44% vs. 81.77%) compared to the best respective values from the baselines. These results suggest that SILAA-SA not only maintains a strong balance between precision and recall, but also outperforms both attention-based and non-attention-based BERT model variants in overall classification performance.

To establish SILAA-SA's effectiveness on the dataset used in our study, we compare its performance with models evaluated in the original FASSILA dataset paper.

Dataset	Study	Model	Acc. %	F1-Score %	Recall %	Precision %
FASSILA [2]	[2]	Arabertv02	83.92	83.03	83.03	78.19
		Arabertv02-KNN	84.22	80.66	82.23	79.29
		Arabertv02-SVM	84.42	80.96	82.22	79.82
		SILAA-SA	**86.81**	**84.46**	**83.74**	**85.44**

Table 6.5: Comparison of SILAA-SA with models from the FASSILA dataset study. The table presents the performance metrics (Accuracy, F1-Score, Recall, and Precision) of SILAA-SA against the models evaluated in the original FASSILA dataset paper on the task of Sentiment Analysis.

As shown in table 6.5 SILAA-SA achieved an accuracy of 86.81%, outperforming the best accuracy achieved by AraBERTv02-SVM (84.42%). Additionally, SILAA-SA demonstrates superior performance across all metrics: it achieves an F1-Score of 84.46%, higher than the best F1-Score achieved by AraBERTv02's 83.03%, indicating a better balance between precision and recall; a recall of 83.74%, surpassing the best recall achieved by AraBERTv02 (83.03%), and a precision of 85.44%, exceeding the best precision achieved by AraBERTv02-SVM (79.82%). These results underscore SILAA-SA's robust performance on the FASSILA dataset, validating its effectiveness for the task and dataset used in our study.

6.3.3 Generalizability of SILAA-SA

In this section, we evaluate the generalizability of SILAA-SA across different languages, tasks, and datasets. We demonstrate its robustness by comparing its performance against state-of-the-

art models on multiple benchmarks, including sentiment analysis and fake news detection tasks. The results highlight SILAA-SA's ability to adapt to diverse linguistic contexts and tasks.

Dataset	Study	Model	Acc. %	F1-Score %	Recall %	Precision %
SANA [133]	[133]	NB	75.00	—	—	—
		SILAA-SA	**88.64**	**88.63**	**88.64**	**88.72**

Table 6.6: Comparison of SILAA-SA with state-of-the-art models on Algerian dialect sentiment analysis task. The table presents the performance metrics (Accuracy, F1-Score, Recall, and Precision) of SILAA-SA against other models on the SANA dataset.

As shown in Table 6.6, SILAA-SA significantly outperforms the previously reported results on the SANA dataset. While the earlier study reported an accuracy of 75% using a Naive Bayes (NB) classifier, SILAA-SA achieves a substantially higher accuracy of 88.64%, along with balanced and consistently high scores across F1-Score, Recall, and Precision. These results reinforce SILAA-SA's ability to generalize effectively beyond the FASSILA dataset.

6.3.3.1 Language Independence

To assess SILAA-SA's language independence, we evaluate its performance on sentiment analysis tasks across Magharebi dialects, Moroccan and Tunisian, and other multi-dialect arabic datasets.

Dataset	Study	Model	Acc. %	F1-Score %	Recall %	Precision %
MAC [82]	[86]	DarijaBERT	90	87.7	87.3	88.1
		Arabertv2	89.6	87.2	87.4	87
	[115]	QARIB	89.96	88.04		
		SILAA-SA	**92.32**	**90.77**	**91.02**	**91.07**
MYC [97]	[86]	Darijabert-arabizi	85.6	85.6	85.6	85.6
		SILAA-SA	**87.25**	**87.22**	**87.40**	**87.18**

Table 6.7: Comparison of SILAA-SA with state-of-the-art models on Moroccan dialect sentiment analysis tasks. The table presents the performance metrics (Accuracy, F1-Score, Recall, and Precision) of SILAA-SA against other models on the MAC and MYC datasets

Table 6.7 compares SILAA-SA with state-of-the-art models on the MAC and MYC datasets, which focus on Moroccan dialect sentiment analysis. On the MAC dataset, SILAA-SA achieves an accuracy of 92.32%, surpassing DarijaBERT (90%), AraBERTv2 (89.6%), and QARIB (89.96%). Similarly, on the MYC dataset, SILAA-SA achieves an accuracy of 87.25%, outperforming DarijaBERT-Arabizi (85.6%), and demonstrating superior performance across all other metrics, including F1-score, recall, and precision, on both datasets.

Dataset	Study	Model	Acc. %	F1-Score %	Recall %	Precision %
TSAC [117]	[118]	TunoBert-CNN	90.8	91.05	90.8	91.3
		CNN-LSTM	81.6	84.36	75.3	95.9
		SILAA-SA	**94.47**	**94.44**	**94.70**	**94.34**

Table 6.8: Comparison of SILAA-SA with state-of-the-art models on Tunisian dialect sentiment analysis task. The table presents the performance metrics (Accuracy, F1-Score, Recall, and Precision) of SILAA-SA against other models on the TSAC dataset.

Table 6.8 presents the comparison of SILAA-SA with state-of-the-art models on the TSAC dataset, which focuses on Tunisian dialect sentiment analysis. SILAA-SA achieves an accuracy of 94.47%, outperforming TunoBERT-CNN (90.8%) and CNN-LSTM (81.6%). This significant improvement demonstrates SILAA-SA's ability to generalize to Tunisian dialect tasks, further validating its language independence.

The results on the MAC, MYC, and TSAC datasets highlight SILAA-SA's ability to adapt to different Arabic dialects, including Moroccan and Tunisian. By consistently outperforming state-of-the-art models across these datasets, SILAA-SA demonstrates its language independence and versatility in processing diverse linguistic contexts.

To further assess SILAA-SA's ability to generalize across dialectal and linguistic variation, we extend our evaluation to the ArSarcasm-v2 dataset [12], a large-scale multi-dialect corpus. While the dataset includes annotations for sarcasm, sentiment, and dialect, our focus in this work is solely on the sentiment analysis task. ArSarcasm-v2 covers a broad spectrum of Arabic varieties, including MSA and regional dialects from Egypt, the Levant, the Gulf, and the Maghreb, offering a challenging benchmark for evaluating robustness across dialects.

Dataset	Study	Model	Acc. %	F1-macro%	W-F1%	F-PN%
ArSarcasm-v2 [12]	[20]	MTL-CNN-LSTM	—	—	—	70.1
	[88]	AraBERT-CNN-BiLSTM	68.40	62.32	—	70.73
	[71]	MTL-ATTINTER	71.07	66.25	—	74.80
	[105]	RAG model	—	—	57	—
	[143]	Ensemble	—	65.70	—	73.92
		SILAA-SA	**71.50**	**66.51**	**70.58**	**75.05**
ArSarcasm-v2 no neutral	[105]	RAG model	—	—	75	—
		SILAA-SA	**87.39**	**84.20**	**87.67**	**—**

Table 6.9: Comparison of SILAA-SA with state-of-the-art models on multi Arabic dialect sentiment analysis task. The table presents the performance metrics (Accuracy, F1-Score, Recall, and Precision) of SILAA-SA against other models on the ArSarcasm-v2 dataset

As illustrated in Table 6.9, SILAA-SA outperforms several state-of-the-art models across a variety of metrics on the ArSarcasm-v2 dataset. Despite the diverse model architectures in prior workincluding MTL-based models, hybrid CNN-BiLSTM networks, and ensemble methods—SILAA-SA consistently achieves the highest values in accuracy (71.50%), F1-macro (66.5%), weighted F1 (70.58%), and the F1-Score for the positive-negative (PN) class (75.05%).

Furthermore, when evaluated on a binary version of the dataset that excludes neutral instances, SILAA-SA achieves a remarkable accuracy of 87.39% and a macro F1-Score of 84.20%, surpassing the best previously reported weighted F1 of 75%.

These improvements are particularly noteworthy given the increased difficulty of handling sentiment analysis in a multi-dialectal dataset, unlike previous experiments that were conducted on more uniform, single-dialect corpora.

6.3.3.2 Task independence

To evaluate SILAA-SA's task independence, we assess its performance on the FASSILA dataset this time for the task of fake news detection.

Dataset	Study	Model	Acc. %	F1-Score %	Recall %	Precision %
FASSILA [2]	[2]	MarBert-KNN	79.08	77.38	76.97	77.80
		MarBert-SVM	79.28	77.55	76.97	78.13
		SILAA-SA	**79.48**	**79.13**	**78.96**	**79.91**

Table 6.10: Comparison of SILAA-SA with models from the FASSILA dataset study. The table presents the performance metrics (Accuracy, F1-Score, Recall, and Precision) of SILAA-SA against the models evaluated in the original FASSILA dataset paper on the task of Fake News.

Table 6.10 compares SILAA-SA with state-of-the-art models evaluated in the original FASSILA dataset paper. SILAA-SA achieves an accuracy of 79.48%, outperforming MarBERT-KNN (79.08%) and MarBERT-SVM (79.28%). Additionally, SILAA-SA demonstrates superior performance in F1-score (79.13%), recall (78.96%), and precision (79.91%) compared to the other models. These results highlight SILAA-SA's ability to generalize beyond sentiment analysis to other NLP tasks, such as fake news detection, while maintaining competitive performance.

To further demonstrate SILAA-SA's task independence, we evaluate its performance on the ANS dataset, which involves fake news detection in Arabic. Table 6.11 compares SILAA-SA with state-of-the-art models on this dataset.

Dataset	Study	Model	Acc. %	F1-Score %	Recall %	Precision %
ANS [108]	[128]	AraBert 2D-CNN	71.42	61.88	52.38	75.6
	[140]	JointBert	66	66	60	62
	[139]	ArabFake	73.24	68.74	68	69.49
		SILAA-SA	**75.00**	**72.64**	**73.53**	**72.13**

Table 6.11: Comparison of SILAA-SA with state-of-the-art models on the ANS dataset for fake news detection. The table presents the performance metrics (Accuracy, F1-Score, Recall, and Precision) of SILAA-SA against other models.

SILAA-SA achieves an accuracy of 75%, outperforming AraBERT 2D-CNN (71.42%), JointBERT (66%), and ArabFake (73.24%). Additionally, SILAA-SA demonstrates superior performance in F1-score (72.64%), recall (73.53%), and precision (72.13%) compared to the other models. These results highlight SILAA-SA's ability to generalize across tasks, showcasing its robustness in handling different tasks such as fake news detection, while also maintaining strong performance across diverse linguistic contexts.

6.4 Ablation Study

To better understand the contribution of each component in SILAA-SA, we conduct a comprehensive ablation study. Table 6.12 presents the results of various model configurations, highlighting the impact of removing or modifying key components on the model's performance.

Model Ablation		Acc. %	F1-Score %	Description
Full Architecture		**86.81**	**84.46**	Complete model with all components.
Unitask		85.25	79.00	Model with all components related to the auxiliary task (MLM) removed. Focuses solely on Sentiment Analysis.
Remove attention		85.03	80.83	Model without the self-attention mechanism.
MoE Ablation	Replace Moe with simple FeedForward	83.33	79.44	MoE replaced with a single feedforward layer.
	Remove Gate Mechanism	85.12	80.41	MoE layer without the gating mechanism.
	Remove Weighted sum and use one expert	85.22	81.14	MoE with only one expert and no weighted sum.
	Remove Experts	84.62	81.35	Model without any MoE experts.

Table 6.12: Results of the component-wise ablation study, showcasing the accuracy and macro F1-score of various model configurations.

- The full architecture, which includes all components (self-attention, multitask learning with MLM, and the MoE layer), achieves the best performance with an accuracy of 86.81% and a macro F1-score of 84.46%. This configuration serves as the baseline for comparison.

- **Unitask Model:** When the auxiliary task (MLM) is removed, resulting in a unitask model focused solely on sentiment analysis, the accuracy drops to 85.25%, and the F1-score decreases to 79.00%. This demonstrates the importance of multitask learning, as the auxiliary task helps the model learn more robust representations that improve performance on the primary task.

- **Remove Self-Attention** Removing the self-attention mechanism reduces the accuracy to 85.03% and the F1-score to 80.83%. This indicates that the self-attention module plays a significant role in capturing contextual dependencies within the input sequence.

- **Impact of MoE Components**

  – **Replace MoE with Simple FeedForward** Replacing the MoE layer with a single feedforward layer results in a significant drop in performance, with accuracy decreas-

ing to 83.33% and F1-score to 79.44%. This highlights the importance of the MoE layer in leveraging multiple experts to capture diverse patterns in the data.

- **Remove Gate Mechanism** Removing the gating mechanism from the MoE layer reduces the accuracy to 85.12% and the F1-score to 80.41%. The gating mechanism is essential for dynamically weighting the contributions of different experts, and its removal degrades the model's ability to adapt to varying input patterns.
- **Remove Weighted Sum and Use One Expert** Using only one expert and removing the weighted sum reduces the accuracy to 85.22% and the F1-score to 81.14%. This shows that the combination of multiple experts and their weighted contributions is critical for achieving optimal performance.
- **Remove Experts** Removing all MoE experts results in an accuracy of 84.62% and an F1-score of 81.35%. This further emphasizes the importance of the MoE layer.

The ablation study demonstrates the importance of each component in SILAA-SA. The full architecture achieves the best performance. Removing any of these components leads to a degradation in performance, highlighting their individual contributions to the model's effectiveness. These findings validate the design choices of SILAA-SA and underscore the importance of each component in achieving state-of-the-art performance.

6.5 Conclusion

In this work, we present SILAA-SA, an innovative multitask learning framework designed to address sentiment analysis in low-resource dialects, with a focus on Algerian Arabic. By leveraging the linguistic similarities between MSA and its dialectssuch as shared lexical, grammatical, and semantic features, SILAA-SA mitigates the resource limitations that hinder NLP applications for under-resourced languages. The framework integrates a Mixture of Experts layer with self-attention mechanisms, enabling it to jointly optimize Algerian Arabic sentiment analysis and MSA masked language modeling. This design captures shared linguistic knowledge, enhancing generalization across datasets, dialects, and tasks. In particular, we attempted to make use of widely available MSA resources in the form of datasets to support and enrich the learning process for low-resource dialects. This chapter addresses Research Questions 5, 6, and 7 by investigating whether MSA data can support learning in low-resource contexts, whether joint learning improves generalization without overfitting, and whether shared representations can transfer across tasks and dialects.

Extensive experiments on Algerian, Tunisian, Moroccan, and other Arabic datasets demonstrate that SILAA-SA achieves state-of-the-art performance in sentiment analysis, outperforming existing models. This highlights the effectiveness of utilizing MSA's extensive resources and the inherent connections between MSA and its dialects to enhance performance in low-resource

settings. Furthermore, SILAA-SA's task-agnostic architecture is validated through competitive results in fake news detection, showcasing its flexibility across diverse NLP tasks.

The results of SILAA-SA emphasize the importance of building on the shared linguistic foundations of high-resource and low-resource languages. By effectively harnessing the structural and semantic overlaps between these languages, SILAA-SA not only advances sentiment analysis in Algerian Arabic but also provides a scalable framework for developing NLP tools for other under-resourced dialects. This work marks a significant stride toward more inclusive and equitable NLP technologies, ensuring that speakers of low-resource languages can fully participate in the digital age. The findings illustrate the potential of multitask learning and shared representations in overcoming the challenges of low-resource NLP, making a meaningful contribution to the field.

CHAPTER 7

GENERAL CONCLUSION

In this thesis, we have thoroughly explored the challenges and solutions related to Arabic dialect identification and sentiment analysis, with a primary focus on low-resource dialects such as Algerian Arabic. This work was motivated by the increasing need to bridge the gap between high-resource languages, like MSA, and underrepresented dialects that lack sufficient linguistic resources. While creating new resources from scratch is an ideal solution for addressing the challenges of low-resource languages, it is a resource-intensive and time-consuming endeavor. Developing large-scale annotated datasets, linguistic tools, and computational infrastructure requires substantial financial, technical, and human resources. This undertaking can take years, if not decades, to fully materialize, making it a daunting task for many communities, particularly those with limited access to funding or technical expertise. Although this strategy has the potential to eventually transform low-resource languages into high-resource ones, the long-term investment required can undermine its feasibility in addressing immediate needs. In the meantime, other methods, such as leveraging existing resources from high-resource languages, offer more immediate solutions for developing NLP capabilities for these languages. Through the development of advanced cross-lingual transfer learning methodologies, we proposed two novel frameworks, WASL-DI for dialect identification and SILAA-SA for sentiment analysis. Additionally, we conducted an in-depth preliminary study comparing traditional machine learning approaches with transfer learning from MSA, which provided complementary insights by exploring two key approaches, the effectiveness of traditional machine learning models in classifying Arabic dialectal text without relying on pretrained linguistic knowledge, and the potential of transfer learning from MSA to enhance model performance. Our disaster classification study demonstrated that traditional machine learning models, despite lacking prior linguistic information, can effectively capture statistical patterns in dialectal text. This finding suggests that

models equipped with prior knowledge although not specifically tailored for dialect identification should be capable of achieving even better performance. Building upon these findings, our investigation into MSA-based transfer learning revealed that leveraging linguistic knowledge from MSA yields results that closely rival specialized models. The promising performance of traditional machine learning models without prior knowledge laid the foundation for this investigation, as it suggested that incorporating MSA a resource-rich and linguistically related source could further enhance performance. This connection highlights how models equipped with even general linguistic knowledge can bridge the resource gap and improve outcomes in dialectal NLP tasks. Although MSA-based pretrained models did not dramatically surpass dialect-specific approaches, the minimal performance gap suggests that linguistic proximity between MSA and Arabic dialects provides a solid foundation for further enhancements. These results support the assumption that integrating resource-rich languages can improve the performance and generalizability of NLP systems in low-resource dialectal contexts.

Our first phase of enhancement leveraged pretrained models to address the challenges identified in the preliminary study, leading to the development of the WASL-DI (Wielding Arabic resources to Support Low-resource Dialect Identification) framework. This framework introduced a hybrid cross-lingual approach that combines CAMeLBERT for contextualized embeddings with FastText subword-level representations. This dual-path architecture effectively integrates deep contextual knowledge with subword-level granularity. Empirical evaluations on the different datasets revealed that WASL-DI consistently outperforms traditional machine learning, deep learning and transformer-based models across multiple dialectal variants. Its superior performance, even in the presence of noisy and incomplete data, demonstrates the model's robustness and its ability to generalize across diverse linguistic contexts. Additionally, the ablation study provided further insights into the contribution of each architectural component, confirming that both CAMeLBERT and FastText embeddings are indispensable for optimal performance.

After leveraging pretrained models in the first phase, we sought to explore another crucial resource, data. The SILAA-SA (Shared Integration of Low-resource Algerian and Arabic for Sentiment Analysis) framework extended the principles of transfer learning to sentiment analysis tasks in dialectal Arabic. By employing a multitask learning paradigm, SILAA-SA jointly optimizes two objectives: dialectal sentiment classification and MSA-based MLM. While MLM serves as an auxiliary task solely for feature sharing, our primary focus remains on improving sentiment classification rather than the MLM performance itself. This design facilitates knowledge sharing between the dialectal and standardized forms of Arabic, enhancing the model's ability to capture sentiment specific patterns across both linguistic varieties. The incorporation of a Mixture of Experts layer further strengthens the model by employing a gating mechanism that assigns weights to different linguistic representations. This dynamic weighting allows the model to prioritize the most relevant features from both MSA and dialectal inputs, improving its ability

to handle the complex nature of dialectal Arabic. Experiments conducted on multiple datasets, including Tunisian and Moroccan Arabic, showcased the model's state-of-the-art performance across a wide range of sentiment analysis tasks. To further validate the independence of the SILAA-SA task design, we tested the framework on fake news datasets. The model not only generalized effectively beyond sentiment analysis but also surpassed previous works, highlighting its adaptability and the strength of the feature-sharing mechanism between MSA and dialectal data.

The contributions of this thesis collectively addressed the seven research questions posed at the outset. The preliminary experiments responded to the first question by demonstrating that models without any prior linguistic knowledge can capture meaningful dialectal patterns, thus motivating the exploration of MSA-based transfer. The WASL-DI framework addressed Research Questions 2, 3, and 4 by evaluating the generalization ability of MSA-trained models, testing their capacity to capture dialect-specific features, and demonstrating the effectiveness of combining contextual and subword-level representations. The SILAA-SA framework answered Research Questions 5, 6, and 7 by directly leveraging MSA data in a multitask setting, showing that joint learning can improve generalization and that shared representations can transfer across tasks and dialects. Together, these contributions provide a comprehensive response to the research questions.

One of the most profound findings of this research is the efficacy of leveraging MSA as a linguistic bridge to address the scarcity of dialect-specific data. The structural and lexical commonalities between MSA and Arabic dialects create an opportunity for effective cross-lingual knowledge transfer. This approach mitigates the limitations posed by insufficient annotated data while maintaining high levels of accuracy across dialectal varieties. The proposed hybrid models not only demonstrate superior performance on dialect identification and sentiment analysis but also highlight the adaptability of our methodologies to broader NLP applications. These frameworks provide a versatile foundation for future research into low-resource language processing, offering scalable solutions that can be tailored to other languages with similar linguistic characteristics.

The contributions of our frameworks extend beyond immediate performance gains by integrating diverse linguistic features, through advanced architectures like the Mixture of Experts and leveraging multitask learning, it provides a robust foundation for addressing low-resource language challenges. The demonstrated adaptability of the our approaches suggests future research could extend these methodologies to other dialect-rich languages or domains where data scarcity is a limiting factor. The demonstrated ability of SILAA-SA to generalize beyond sentiment analysis to tasks like fake news detection highlights the potential for cross-domain transfer learning, opening opportunities to expand the frameworks to other complex NLP tasks, while maintaining robust performance across domains. The principles of cross-lingual transfer learning, as demonstrated in this thesis, are not limited to Arabic dialects but could be applied to

other low-resource languages with dialectal variations or limited annotated datasets. Despite these advances, the work presented in this thesis is not without limitations. The use of only text-based input restricts the applicability of the models to multimodal scenarios, and further evaluation on code-switched and mixed-script text is necessary. Additionally, while MSA serves as a useful high-resource proxy, dialects with greater divergence from MSA may require more specialized adaptation. Furthermore, future work could first explore integrating multimodal learning techniques combining textual, audio, and visual data to enhance the model's robustness and generalizability, enabling it to process diverse linguistic inputs more effectively, additionaly, refining model architectures to better handle code-switched and mixed-script data could address a persistent challenge in dialectal NLP, finally, exploring zero-shot or few-shot transfer techniques would allow rapid adaptation to dialects with even less available data.

The research presented in this thesis has already led to several peer-reviewed publications and conference presentations, contributing to the growing body of work in Arabic dialect processing and cross-lingual NLP.

In conclusion, this thesis makes significant contributions to the field of Arabic Natural Language Processing by advancing the state-of-the-art in dialect identification and sentiment analysis. Through the development of the WASL-DI and SILAA-SA frameworks, we have demonstrated that cross-lingual transfer learning from MSA is a powerful strategy for improving model performance in low-resource dialectal contexts. By leveraging resources from high-resource language varieties, our work takes a crucial step toward bridging the gap between resource-rich and resource-scarce dialects and languages, offering a practical solution to the persistent challenge of low-resource settings. These findings not only establish new benchmarks for Arabic dialect tasks but also provide a scalable methodology that can be adapted to other linguistic domains. This work lays a strong foundation for future innovations aimed at fostering greater inclusivity and accessibility in language technologies. As NLP continues to evolve, it is imperative to prioritize research that empowers all linguistic communities, ensuring that the benefits of AI-driven language technologies are shared equitably across the globe.

BIBLIOGRAPHY

[1] A. ABDAOUI, M. BERRIMI, M. OUSSALAH, AND A. MOUSSAOUI, *Dziribert: a pre-trained language model for the algerian dialect*, 2021.

[2] A. ABDEDAIEM, A. H. DAHOU, M. A. CHERAGUI, AND B. MATHIAK, *Fassila: A corpus for algerian dialect fake news detection and sentiment analysis*, Procedia Computer Science, 244 (2024), pp. 397–407.

[3] A. ABDELALI, S. HASSAN, H. MUBARAK, K. DARWISH, AND Y. SAMIH, *Pre-training bert on arabic tweets: Practical considerations*, (2021).

[4] A. ABDELALI, H. MUBARAK, Y. SAMIH, S. HASSAN, AND K. DARWISH, *QADI: Arabic dialect identification in the wild*, in Proceedings of the Sixth Arabic Natural Language Processing Workshop, Association for Computational Linguistics, Kiev, Ukraine, 2021, pp. 1–10.

[5] M. ABDELAZIZ, *How many countries speak arabic around the world?* `https://tarjama.com/how-many-countries-that-speak-arabic-around-the-world/`, 2022.
Tarjama, Retrieved June 18, 2024.

[6] D. A. ABDO, *Stress and Arabic phonology*, University of Illinois at Urbana-Champaign, 1969.

[7] M. ABDUL-MAGEED, A. ELMADANY, C. ZHANG, E. NAGOUDI, H. BOUAMOR, AND N. HABASH, *NADI 2023: The Fourth Nuanced Arabic Dialect Identification Shared Task*, in Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023), 2023.

[8] M. ABDUL-MAGEED, A. KELEG, A. ELMADANY, C. ZHANG, I. HAMED, W. MAGDY, H. BOUAMOR, AND N. HABASH, *Nadi 2024: The fifth nuanced arabic dialect identification shared task*, in Proceedings of the Eighth Arabic Natural Language Processing Workshop (WANLP), N. Habash, H. Bouamor, R. Eskander, N. Tomeh, I. Farha, A. Abdelali, S. Touileb, I. Hamed, Y. Onaizan, B. Alhafni, W. Antoun, S. Khalifa, H. Haddad, I. Zitouni, B. AlKhamissi, and R. Almatham, eds., 2024.

[9] M. ABDUL-MAGEED, C. ZHANG, H. BOUAMOR, AND N. HABASH, *Nadi 2020: The first nuanced arabic dialect identification shared task*, in Proceedings of the Fifth Arabic Natural Language Processing Workshop, I. Zitouni, M. Abdul-Mageed, H. Bouamor, F. Bougares, M. El-Haj, N. Tomeh, and W. Zaghouani, eds., Association for Computational Linguistics, 2020, pp. 97–110.

[10] M. ABDUL-MAGEED, C. ZHANG, A. ELMADANY, H. BOUAMOR, AND N. HABASH, *Nadi 2021: The second nuanced arabic dialect identification shared task*, in Proceedings of the Sixth Arabic Natural Language Processing Workshop, N. Habash, H. Bouamor, H. Hajj, W. Magdy, W. Zaghouani, F. Bougares, N. Tomeh, I. Farha, and S. Touileb, eds., Association for Computational Linguistics, 2021, pp. 244–259.

[11] M. ABDUL-MAGEED, C. ZHANG, A. ELMADANY, H. BOUAMOR, AND N. HABASH, *Nadi 2022: The third nuanced Arabic dialect identification shared task*, in Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), H. Bouamor, H. Al-Khalifa, K. Darwish, O. Rambow, F. Bougares, A. Abdelali, N. Tomeh, S. Khalifa, and W. Zaghouani, eds., Abu Dhabi, United Arab Emirates (Hybrid), dec 2022, Association for Computational Linguistics, pp. 85–97.

[12] I. ABU FARHA, W. ZAGHOUANI, AND W. MAGDY, *Overview of the wanlp 2021 shared task on sarcasm and sentiment detection in arabic*, in Proceedings of the Sixth Arabic Natural Language Processing Workshop, April 2021.

[13] J. ACHIAM, S. ADLER, S. AGARWAL, L. AHMAD, I. AKKAYA, F. L. ALEMAN, D. ALMEIDA, J. ALTENSCHMIDT, S. ALTMAN, S. ANADKAT, ET AL., *Gpt-4 technical report*, arXiv preprint arXiv:2303.08774, (2023).

[14] S. ADEL AND N. ELMADANY, *Isl-aast at nadi 2023 shared task: Enhancing arabic dialect identification in the era of globalization and technological progress*, in Proceedings of ArabicNLP 2023, H. Sawaf, S. El-Beltagy, W. Zaghouani, W. Magdy, A. Abdelali, N. Tomeh, I. Abu Farha, N. Habash, S. Khalifa, A. Keleg, H. Haddad, I. Zitouni, K. Mrini, and R. Almatham, eds., Singapore (Hybrid), Dec. 2023, Association for Computational Linguistics, pp. 631–636.

[15] S. AHMAD, S. SAQIB, AND A. SYED, *Cnn and lstm based hybrid deep learning model for sentiment analysis on arabic text reviews*, Mehran University Research Journal of Engineering and Technology, 43 (2024), pp. 183–194.

[16] N. AL-TWAIRESH, H. AL-KHALIFA, AND A. AL-SALMAN, *AraSenTi: Large-scale Twitter-specific Arabic sentiment lexicons*, in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), K. Erk and N. A.

Smith, eds., Berlin, Germany, aug 2016, Association for Computational Linguistics, pp. 697–705.

[17] F. ALAM, F. OFLI, AND M. IMRAN, *Crisisdps: Crisis data processing services*, in ISCRAM, 2019.

[18] A. M. ALAYBA, V. PALADE, M. ENGLAND, AND R. IQBAL, *Arabic language sentiment analysis on health services*, in 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), 2017, pp. 114–118.

[19] A. ALHARBI AND M. LEE, *Classifying arabic crisis tweets using data selection and pre-trained language models*, in Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Quran QA and Fine-Grained Hate Speech Detection, 2022, pp. 71–78.

[20] A. I. ALHARBI AND M. LEE, *Multi-task learning using a combination of contextualised and static word embeddings for Arabic sarcasm detection and sentiment analysis*, in Proceedings of the Sixth Arabic Natural Language Processing Workshop, N. Habash, H. Bouamor, H. Hajj, W. Magdy, W. Zaghouani, F. Bougares, N. Tomeh, I. Abu Farha, and S. Touileb, eds., Kyiv, Ukraine (Virtual), Apr. 2021, Association for Computational Linguistics, pp. 318–322.

[21] B. ALHARBI, H. ALAMRO, M. ALSHEHRI, Z. KHAYYAT, M. KALKATAWI, I. I. JABER, AND X. ZHANG, *Asad: A twitter-based benchmark arabic sentiment analysis dataset*, 2021.

[22] K. M. ALHAWITI, *Natural language processing and its use in education*, International Journal of Advanced Computer Science and Applications, 5 (2014).

[23] A. ALIWY, H. TAHER, AND Z. ABOALTAHEEN, *Arabic dialects identification for all Arabic countries*, in Proceedings of the Fifth Arabic Natural Language Processing Workshop, I. Zitouni, M. Abdul-Mageed, H. Bouamor, F. Bougares, M. El-Haj, N. Tomeh, and W. Zaghouani, eds., Barcelona, Spain (Online), Dec. 2020, Association for Computational Linguistics, pp. 302–307.

[24] B. ALKHAMISSI, M. GABR, M. ELNOKRASHY, AND K. ESSAM, *Adapting marbert for improved arabic dialect identification: Submission to the nadi 2021 shared task*, in Proceedings of the Sixth Arabic Natural Language Processing Workshop, N. Habash, H. Bouamor, H. Hajj, W. Magdy, W. Zaghouani, F. Bougares, N. Tomeh, I. Abu Farha, and S. Touileb, eds., Kyiv, Ukraine (Virtual), Apr. 2021, Association for Computational Linguistics, pp. 260–264.

[25] B. ALMUHAYA, B. SAHA, M. KAUR, M. A. BAZEL, AND R. MOHAMMED, *Comparative analysis of machine learning algorithms for arabic sentiment analysis on imbalanced*

social media data, in 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), 2024, pp. 1362–1367.

[26] L. ALMUQREN AND A. CRISTEA, *Aracust: a saudi telecom tweets corpus for sentiment analysis*, PeerJ Comput Sci, 7 (2021), p. e510.

[27] K. M. ALOMARI, H. M. ELSHERIF, AND K. SHAALAN, *Arabic tweets sentimental analysis using machine learning*, in International conference on industrial, engineering and other applications of applied intelligent systems, Springer, 2017, pp. 602–610.

[28] W. ALOSAIMI, H. SALEH, A. A. HAMZAH, N. EL-RASHIDY, A. ALHARB, A. ELARABY, AND S. MOSTAFA, *Arabbert-lstm: improving arabic sentiment analysis based on transformer model and long short-term memory*, Frontiers in Artificial Intelligence, 7 (2024).

[29] O. ALSEMAREE, A. S. ALAM, S. S. GILL, AND S. UHLIG, *Sentiment analysis of arabic social media texts: A machine learning approach to deciphering customer perceptions*, Heliyon, 10 (2024), p. e27863.

[30] H. ALSHAMMARI, A. EL-SAYED, AND K. ELLEITHY, *Ai-generated text detector for arabic language using encoder-based transformer architecture*, Big Data and Cognitive Computing, 8 (2024).

[31] N. ALSHENAIFI AND A. AZMI, *Arabic dialect identification using machine learning and transformer-based models: Submission to the nadi 2022 shared task*, in Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), H. Bouamor, H. Al-Khalifa, K. Darwish, O. Rambow, F. Bougares, A. Abdelali, N. Tomeh, S. Khalifa, and W. Zaghouani, eds., Abu Dhabi, United Arab Emirates (Hybrid), Dec. 2022, Association for Computational Linguistics, pp. 464–467.

[32] A. ALSUWAYLIMI, *Arabic dialect dataset*.
`https://www.kaggle.com/datasets/amjadalsuwaylimi/arabic-dialect-dataset/data`.
Retrieved 10/09/2024.

[33] A. A. ALSUWAYLIMI, *Arabic dialect identification in social media: A hybrid model with transformer models and bilstm*, Heliyon, 10 (2024), p. e36280.

[34] A. ALWEHAIBI, M. BIKDASH, M. ALBOGMI, AND K. ROY, *A study of the performance of embedding methods for arabic short-text sentiment analysis using deep learning approaches*, Journal of King Saud University - Computer and Information Sciences, 34 (2022), pp. 6140–6149.

[35] M. A. ALY AND A. F. ATIYA, *Labr: A large scale arabic book reviews dataset*, ArXiv, abs/1411.6718 (2013).

[36] S. N. ALYAMI AND S. O. OLATUNJI, *Application of support vector machine for arabic sentiment classification using twitter-based dataset*, Journal of Information & Knowledge Management, 19 (2020), p. 2040018.

[37] W. ANTOUN, F. BALY, AND H. HAJJ, *Arabert: Transformer-based model for arabic language understanding*, in Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, 2020, pp. 9–15.

[38] J. E. AOUN, E. BENMAMOUN, AND L. CHOUEIRI, *The syntax of Arabic*, Cambridge University Press, 2009.

[39] ATDFS, *Large arabic twitter data for sentiment analysis*, 2019.

[40] J. ATTIEH AND F. HASSAN, *Arabic dialect identification and sentiment classification using transformer-based models*, in Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), H. Bouamor, H. Al-Khalifa, K. Darwish, O. Rambow, F. Bougares, A. Abdelali, N. Tomeh, S. Khalifa, and W. Zaghouani, eds., Abu Dhabi, United Arab Emirates (Hybrid), Dec. 2022, Association for Computational Linguistics, pp. 485–490.

[41] D. BAHDANAU, K. CHO, AND Y. BENGIO, *Neural machine translation by jointly learning to align and translate*, CoRR, abs/1409.0473 (2014).

[42] N. BAIMUKAN, H. BOUAMOR, AND N. HABASH, *Hierarchical aggregation of dialectal data for Arabic dialect identification*, in Proceedings of the Thirteenth Language Resources and Evaluation Conference, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, eds., Marseille, France, jun 2022, European Language Resources Association, pp. 4586–4596.

[43] D. M. A. BAKHIT, L. NDERU, AND A. NGUNYI, *A hybrid neural network model based on transfer learning for arabic sentiment analysis of customer satisfaction*, Engineering Reports, 6 (2024), p. e12874.

[44] S. BALLOUT, *People of arab heritage*, in Handbook for Culturally Competent Care, Springer, 2024, pp. 97–137.

[45] R. BALY, A. KHADDAJ, H. M. HAJJ, W. EL-HAJJ, AND K. B. SHABAN, *Arsentd-lev: A multi-topic corpus for target-based sentiment analysis in arabic levantine tweets*, CoRR, abs/1906.01830 (2019).

[46] M. BASU, A. SHANDILYA, P. KHOSLA, K. GHOSH, AND S. GHOSH, *Extracting resource needs and availabilities from microblogs for aiding post-disaster relief operations*, IEEE Transactions on Computational Social Systems, 6 (2019), pp. 604–618.

[47] G. BAYRAK AND A. M. ISSIFU, *Domain-adapted bert-based models for nuanced arabic dialect identification and tweet sentiment analysis*, in Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), H. Bouamor, H. Al-Khalifa, K. Darwish, O. Rambow, F. Bougares, A. Abdelali, N. Tomeh, S. Khalifa, and W. Zaghouani, eds., Abu Dhabi, United Arab Emirates (Hybrid), Dec. 2022, Association for Computational Linguistics, pp. 425–430.

[48] A. BELTAGY, A. ABOUELENIN, AND O. ELSHERIEF, *Arabic dialect identification using bert-based domain adaptation*, in Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, 2020, pp. 35–42.

[49] E. M. BENDER, T. GEBRU, A. MCMILLAN-MAJOR, AND S. SHMITCHELL, *On the dangers of stochastic parrots: Can language models be too big?* , in Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, New York, NY, USA, 2021, Association for Computing Machinery, p. 610623.

[50] Z. BENMOUNAH AND A. BOULESNANE, *Algerian dialect*. `https://www.mendeley.com/data/zzwg3nnhsz.2`. doi:10.17632/zzwg3nnhsz.2.

[51] M. BERRIMI, M. OUSSALAH, A. MOUSSAOUI, AND M. SAIDI, *Attention mechanism architecture for arabic sentiment analysis*, ACM Trans. Asian Low-Resour. Lang. Inf. Process., 22 (2023).

[52] P. BOJANOWSKI, E. GRAVE, A. JOULIN, AND T. MIKOLOV, *Enriching word vectors with subword information*, Transactions of the Association for Computational Linguistics, 5 (2017), pp. 135–146.

[53] H. BOUAMOR, N. HABASH, M. SALAMEH, W. ZAGHOUANI, O. RAMBOW, D. ABDULRAHIM, O. OBEID, S. KHALIFA, F. ERYANI, A. ERDMANN, AND K. OFLAZER, *The madar Arabic dialect corpus and lexicon*, in Proceedings of the Language Resources and Evaluation Conference (LREC), 2018.

[54] H. BOUAMOR, S. HASSAN, AND N. HABASH, *The madar shared task on arabic fine-grained dialect identification*, in Proceedings of the Fourth Arabic Natural Language Processing Workshop, Association for Computational Linguistics, 2019, pp. 199–207.

[55] A. BOUZENZANA, *Algerian car market comments sentiment dataset*. `https://www.kaggle.com/datasets/abdeldjalilbouz/algerian-car-market-comments-sentim` Retrieved 24.03.2024.

[56] L. BREIMAN, *Bagging predictors*, Machine learning, 24 (1996), pp. 123–140.

[57] ——, *Random forests*, Machine learning, 45 (2001), pp. 5–32.

[58] L. BREIMAN, J. FRIEDMAN, R. A. OLSHEN, AND C. J. STONE, *Classification and Regression Trees*, Chapman and Hall/CRC, 1st ed., 1984.

[59] T. BUCKWALTER, *Arabic transliteration*, 2001.
Retrieved November 4, 2024.

[60] R. CARUANA, *Multitask learning*, Machine Learning, 28 (1997), pp. 41–75.

[61] T. CHEN AND C. GUESTRIN, *XGBoost: A Scalable Tree Boosting System*, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), San Francisco, California, USA, 2016, ACM, pp. 785–794.

[62] H. CHOUIKHI, H. CHNITER, AND F. JARRAY, *Arabic sentiment analysis using bert model*, in International conference on computational collective intelligence, Springer, 2021, pp. 621–632.

[63] J. CHUNG, ÇAGLAR GÜLÇEHRE, K. CHO, AND Y. BENGIO, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, ArXiv, abs/1412.3555 (2014).

[64] C. CORTES AND V. VAPNIK, *Support-vector networks*, Machine learning, 20 (1995), pp. 273–297.

[65] C. CRAWL, *Common crawl corpus*, Online at http://commoncrawl. org, (2019).

[66] G. DE FRANCONY, V. GUICHARD, P. JOSHI, H. AFLI, AND A. BOUCHEKIF, *Hierarchical deep learning for Arabic dialect identification*, in Proceedings of the Fourth Arabic Natural Language Processing Workshop, W. El-Hajj, L. H. Belguith, F. Bougares, W. Magdy, I. Zitouni, N. Tomeh, M. El-Haj, and W. Zaghouani, eds., Florence, Italy, Aug. 2019, Association for Computational Linguistics, pp. 249–253.

[67] J. DEVLIN, M.-W. CHANG, K. LEE, AND K. TOUTANOVA, *BERT: Pre-training of deep bidirectional transformers for language understanding*, in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), J. Burstein, C. Doran, and T. Solorio, eds., Minneapolis, Minnesota, jun 2019, Association for Computational Linguistics, pp. 4171–4186.

[68] T. G. DIETTERICH, *Ensemble methods in machine learning*, International workshop on multiple classifier systems, (2000), pp. 1–15.

[69] D. EBERHARD, G. SIMONS, AND C. FENNIG, eds., *Ethnologue: Languages of the World*, SIL International, Dallas, Texas, twenty-seventh ed., 2024.
Online version: `http://www.ethnologue.com`.

[70] M. EL-HAJ, P. RAYSON, AND M. ABOELEZZ, *Arabic dialect identification in the context of bivalency and code-switching*, in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, eds., Miyazaki, Japan, May 2018, European Language Resources Association (ELRA).

[71] A. EL MAHDAOUY, A. EL MEKKI, K. ESSEFAR, N. EL MAMOUN, I. BERRADA, AND A. KHOUMSI, *Deep multi-task model for sarcasm detection and sentiment analysis in Arabic language*, in Proceedings of the Sixth Arabic Natural Language Processing Workshop, N. Habash, H. Bouamor, H. Hajj, W. Magdy, W. Zaghouani, F. Bougares, N. Tomeh, I. Abu Farha, and S. Touileb, eds., Kyiv, Ukraine (Virtual), apr 2021, Association for Computational Linguistics, pp. 334–339.

[72] A. EL MEKKI, A. EL MAHDAOUY, K. ESSEFAR, N. EL MAMOUN, I. BERRADA, AND A. KHOUMSI, *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, N. Habash, H. Bouamor, H. Hajj, W. Magdy, W. Zaghouani, F. Bougares, N. Tomeh, I. Abu Farha, and S. Touileb, eds., Kyiv, Ukraine (Virtual), Apr. 2021, Association for Computational Linguistics, pp. 271–275.

[73] T. A. EL-SADANY AND M. A. HASHISH, *An arabic morphological system*, IBM Systems Journal, 28 (1989), pp. 600–612.

[74] M. ELARABY AND A. ZAHRAN, *A character level convolutional BiLSTM for Arabic dialect identification*, in Proceedings of the Fourth Arabic Natural Language Processing Workshop, W. El-Hajj, L. H. Belguith, F. Bougares, W. Magdy, I. Zitouni, N. Tomeh, M. El-Haj, and W. Zaghouani, eds., Florence, Italy, Aug. 2019, Association for Computational Linguistics, pp. 274–278.

[75] M. ELKAREF, M. MOSES, S. TANAKA, J. BARRY, AND G. MEL, *Nlpeople at nadi 2023 shared task: Arabic dialect identification with augmented context and multi-stage tuning*, in Proceedings of ArabicNLP 2023, H. Sawaf, S. El-Beltagy, W. Zaghouani, W. Magdy, A. Abdelali, N. Tomeh, I. Abu Farha, N. Habash, S. Khalifa, A. Keleg, H. Haddad, I. Zitouni, K. Mrini, and R. Almatham, eds., Singapore (Hybrid), Dec. 2023, Association for Computational Linguistics, pp. 642–646.

[76] A. ELNAGAR AND O. EINEA, *Brad 1.0: Book reviews in arabic dataset*, in 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), 2016, pp. 1–8.

[77] A. ELNAGAR, Y. KHALIFA, AND A. EINEA, *Hotel arabic-reviews dataset construction for sentiment analysis applications*, 2018.

[78] Y. FARES, Z. EL-ZANATY, K. ABDEL-SALAM, M. EZZELDIN, A. MOHAMED, K. EL-AWAAD, AND M. TORKI, *Arabic dialect identification with deep learning and hybrid frequency-based features*, in Proceedings of the Fourth Arabic Natural Language Processing Workshop, W. El-Hajj, L. H. Belguith, F. Bougares, W. Magdy, I. Zitouni, N. Tomeh, M. El-Haj, and W. Zaghouani, eds., Florence, Italy, Aug. 2019, Association for Computational Linguistics, pp. 224–228.

[79] C. A. FERGUSON, *Diglossia*, Word, 15 (1959), pp. 325–340.

[80] H. FOUADI, H. EL MOUBTAHIJ, H. LAMTOUGUI, AND A. YAHYAOUY, *Bert-based models for classifying multi-dialect arabic texts*, IAES International Journal of Artificial Intelligence (IJ-AI), 13 (2024), p. 3437.

[81] Y. FREUND AND R. E. SCHAPIRE, *A decision-theoretic generalization of on-line learning and an application to boosting*, Journal of Computer and System Sciences, 55 (1997), pp. 119–139.

[82] M. GAROUANI AND J. KHARROUBI, *Mac: an open and free moroccan arabic corpus for sentiment analysis*, in The Proceedings of the International Conference on Smart City Applications, Springer, 2021, pp. 849–858.

[83] A. GOKASLAN AND V. COHEN, *Openwebtext corpus*. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

[84] M. HADWAN, M. A. AL-HAGERY, M. AL-SAREM, AND F. SAEED, *Arabic sentiment analysis of users opinions of governmental mobile applications*, Computers, Materials and Continua, 72 (2022), pp. 4675–4689.

[85] J. HAN, M. KAMBER, AND J. PEI, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2012.

[86] M. HANNANI, A. SOUDI, AND K. VAN LAERHOVEN, *Assessing the performance of ChatGPT-4, fine-tuned BERT and traditional ML models on Moroccan Arabic sentiment analysis*, in Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities, M. Hämäläinen, E. Öhman, S. Miyagawa, K. Alnajjar, and Y. Bizzoni, eds., Miami, USA, Nov. 2024, Association for Computational Linguistics, pp. 489–498.

[87] J. HEATON, I. GOODFELLOW, Y. BENGIO, AND A. COURVILLE, *Deep learning*, Genetic Programming and Evolvable Machines, 19 (2018), pp. 305–307.

[88] A. HENGLE, A. KSHIRSAGAR, S. DESAI, AND M. MARATHE, *Combining context-free and contextualized representations for Arabic sarcasm detection and sentiment identification*, in Proceedings of the Sixth Arabic Natural Language Processing Workshop, N. Habash, H. Bouamor, H. Hajj, W. Magdy, W. Zaghouani, F. Bougares, N. Tomeh, I. Abu Farha, and S. Touileb, eds., Kyiv, Ukraine (Virtual), apr 2021, Association for Computational Linguistics, pp. 357–363.

[89] S. HOCHREITER AND J. SCHMIDHUBER, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.

[90] M. HUMAYUN, H. YASSIN, J. SHUJA, AND ET AL., *A transformer fine-tuning strategy for text dialect identification*, Neural Computing and Applications, 35 (2023), pp. 6115–6124.

[91] G. INOUE, B. ALHAFNI, N. BAIMUKAN, H. BOUAMOR, AND N. HABASH, *The interplay of variant, size, and task type in arabic pre-trained language models*, (2021).

[92] O. G. IROJU AND J. O. OLALEKE, *A systematic review of natural language processing in healthcare*, International Journal of Information Technology and Computer Science, 8 (2015), pp. 44–50.

[93] E. ISSA, M. ALSHAKHORI1, R. AL-BAHRANI, AND G. HAHN-POWELL, *Country-level arabic dialect identification using rnns with and without linguistic features*, in Proceedings of the Sixth Arabic Natural Language Processing Workshop, N. Habash, H. Bouamor, H. Hajj, W. Magdy, W. Zaghouani, F. Bougares, N. Tomeh, I. Abu Farha, and S. Touileb, eds., Kyiv, Ukraine (Virtual), Apr. 2021, Association for Computational Linguistics, pp. 276–281.

[94] M. ISSIGHID, *Algerian corpus*.
`https://www.kaggle.com/datasets/massinissaissighid/algerian-corpus-algerian-dataset`.
Retrieved 24.03.2024.

[95] R. A. JACOBS, M. I. JORDAN, S. J. NOWLAN, AND G. E. HINTON, *Adaptive mixtures of local experts*, Neural Computation, 3 (1991), pp. 79–87.

[96] H. JAUHIAINEN, K. ALNAJJAR, T. HONKELA, AND K. LINDÉN, *Optimizing naive bayes for arabic dialect identification*, in Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), 2022, pp. 366–375.

[97] M. JBEL, M. JABRANE, I. HAFIDI, AND A. METRANE, *Sentiment analysis dataset in moroccan dialect: bridging the gap between arabic and latin scripted dialect*, Language Resources and Evaluation, (2024), pp. 1–30.

[98] M. I. JORDAN AND R. A. JACOBS, *Hierarchical mixtures of experts and the em algorithm*, Neural Computation, 6 (1994), pp. 181–214.

[99] A. JOSHI, R. DABRE, D. KANOJIA, Z. LI, H. ZHAN, G. HAFFARI, AND D. DIPPOLD, *Natural language processing for dialects of a language: A survey*, ACM Comput. Surv., (2025).

[100] P. JOSHI, S. SANTY, A. BUDHIRAJA, K. BALI, AND M. CHOUDHURY, *The state and fate of linguistic diversity and inclusion in the NLP world*, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds., Online, jul 2020, Association for Computational Linguistics, pp. 6282–6293.

[101] P. JOSHI, S. SANTY, A. BUDHIRAJA, K. BALI, AND M. CHOUDHURY, *The state and fate of linguistic diversity and inclusion in the nlp world*, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020, pp. 6282–6293.

[102] V. KANJIRANGAT, T. SAMARDZIC, L. DOLAMIC, AND F. RINALDI, *Nlp_di at nadi 2024 shared task: Multi-label arabic dialect classifications with an unsupervised cross-encoder*, in Proceedings of The Second Arabic Natural Language Processing Conference, N. Habash, H. Bouamor, R. Eskander, N. Tomeh, I. Abu Farha, A. Abdelali, S. Touileb, I. Hamed, Y. Onaizan, B. Alhafni, W. Antoun, S. Khalifa, H. Haddad, I. Zitouni, B. AlKhamissi, R. Almatham, and K. Mrini, eds., Bangkok, Thailand, Aug. 2024, Association for Computational Linguistics, pp. 742–747.

[103] A. KAROUI, F. GHARBI, R. KAMMOUN, I. LAOUIRINE, AND F. BOUGARES, *Elyadata at nadi 2024 shared task: Arabic dialect identification with similarity-induced mono-to-multi label transformation*, in Proceedings of The Second Arabic Natural Language Processing Conference, N. Habash, H. Bouamor, R. Eskander, N. Tomeh, I. Abu Farha, A. Abdelali, S. Touileb, I. Hamed, Y. Onaizan, B. Alhafni, W. Antoun, S. Khalifa, H. Haddad, I. Zitouni, B. AlKhamissi, R. Almatham, and K. Mrini, eds., Bangkok, Thailand, Aug. 2024, Association for Computational Linguistics, pp. 758–763.

[104] A. KASEB AND M. FAROUK, *Active learning for arabic sentiment analysis*, Alexandria Engineering Journal, 77 (2023), pp. 177–187.

[105] S. KHALED, E. H. MOHAMED, AND W. MEDHAT, *Evaluating large language models for arabic sentiment analysis: A comparative study using retrieval-augmented generation*, Procedia Computer Science, 244 (2024), pp. 363–370.
6th International Conference on AI in Computational Linguistics.

[106] M. KHAN, K. ULLAH, Y. ALHARBI, A. ALFERAIDI, T. ALHARBI, K. YADAV, N. ALSHARABI, AND A. AHMAD, *Understanding the research challenges in low-resource language and*

linking bilingual news articles in multilingual news archive, Applied Sciences, 13 (2023).

[107] A. KHOOLI, *Arabic 100k reviews*.
`https://www.kaggle.com/datasets/abedkhooli/arabic-100k-reviews`.
Retrieved 23.03.2024.

[108] J. KHOUJA, *Stance prediction and claim verification: An arabic perspective*, arXiv preprint arXiv:2005.10410, (2020).

[109] A. KIROUANE, *Algerian-darija*.
`https://huggingface.co/datasets/ayoubkirouane/Algerian-Darija`.
Retrieved 15.07.2024.

[110] S. KUNDU, P. K. SRIJITH, AND M. S. DESARKAR, *Classification of short-texts generated during disasters: A deep neural network based approach*, in 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, Spain, 2018, pp. 790–793.

[111] W. LABOV, *Sociolinguistic Patterns*, Conduct and Communication, 4, University of Pennsylvania Press, Philadelphia, 1972.
Reviewed by E. C. Traugott in *Language in Society*, 1975, 4(1):89–107. doi:10.1017/S0047404500004528.

[112] Y. LECUN, L. BOTTOU, Y. BENGIO, AND P. HAFFNER, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.

[113] C. MANNING, *Foundations of statistical natural language processing*, The MIT Press, 1999.

[114] M. MANSOUR, M. TOHAMY, Z. EZZAT, AND M. TORKI, *Arabic dialect identification using bert fine-tuning*, in Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, 2020, pp. 25–34.

[115] Y. MATRANE, F. BENABBOU, AND Z. ELLAKY, *Enhancing moroccan dialect sentiment analysis through optimized preprocessing and transfer learning techniques*, IEEE Access, 12 (2024), pp. 187756–187777.

[116] Q. MCNEMAR, *Note on the sampling error of the difference between correlated proportions or percentages*, Psychometrika, 12 (1947), pp. 153–157.

[117] S. MDHAFFAR, F. BOUGARES, Y. ESTEVE, AND L. HADRICH-BELGUITH, *Sentiment analysis of tunisian dialects: Linguistic ressources and experiments*, in Third Arabic natural language processing workshop (WANLP), 2017, pp. 55–61.

[118] S. MECHTI, R. FAIZ, N. KHOUFI, S. ANTIT, AND M. KRICHEN, *Sentiment analysis: Effect of combining bert as an embedding technique with cnn model for tunisian dialect*, in International Conference on Information and Knowledge Systems, Springer, 2023, pp. 309–320.

[119] K. MEFTOUH, N. BOUCHEMAL, AND K. SMAÏLI, *A study of a non-resourced language: an algerian dialect*, in Proceedings of the Spoken Language Technologies for Under-resourced Languages (SLTU), 2012, pp. 125–132.

[120] M. MHAMED, R. SUTCLIFFE, X. SUN, J. FENG, E. ALMEKHLAFI, AND E. A. RETTA, *Improving arabic sentiment analysis using cnn-based architectures and text preprocessing*, Computational Intelligence and Neuroscience, 2021 (2021), p. 5538791.

[121] S. MIHI, B. A. B. ALI, I. E. BAZI, S. AREZKI, AND N. LAACHFOUBI, *Mstd: Moroccan sentiment twitter dataset*, International Journal of Advanced Computer Science and Applications, 11 (2020).

[122] T. MIKOLOV, K. CHEN, G. CORRADO, AND J. DEAN, *Efficient estimation of word representations in vector space*, 2013.

[123] P. MISHRA AND V. MUJADIA, *Arabic dialect identification for travel and Twitter text*, in Proceedings of the Fourth Arabic Natural Language Processing Workshop, W. El-Hajj, L. H. Belguith, F. Bougares, W. Magdy, I. Zitouni, N. Tomeh, M. El-Haj, and W. Zaghouani, eds., Florence, Italy, aug 2019, Association for Computational Linguistics, pp. 234–238.

[124] A. MOHAMMED, Z. JIANGBIN, AND A. MURTADHA, *A three stage neural model for arabic dialect identification*, Computer Speech & Language, 80 (2023), p. 101488.

[125] M. NABIL, M. ALY, AND A. ATIYA, *ASTD: Arabic sentiment tweets dataset*, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, L. Màrquez, C. Callison-Burch, and J. Su, eds., Lisbon, Portugal, sep 2015, Association for Computational Linguistics, pp. 2515–2519.

[126] H. NAYEL, A. HASSAN, M. SOBHI, AND A. EL-SAWY, *Machine learning-based approach for arabic dialect identification*, in Proceedings of the Sixth Arabic Natural Language Processing Workshop, N. Habash, H. Bouamor, H. Hajj, W. Magdy, W. Zaghouani, F. Bougares, N. Tomeh, I. Abu Farha, and S. Touileb, eds., Kyiv, Ukraine (Virtual), Apr. 2021, Association for Computational Linguistics, pp. 287–290.

[127] A. F. A. NWESRI, N. A. S. SHINBIR, AND H. EBRAHEM, *Uot at nadi 2023 shared task: Automatic arabic dialect identification is made possible*, in Proceedings of ArabicNLP 2023, H. Sawaf, S. El-Beltagy, W. Zaghouani, W. Magdy, A. Abdelali, N. Tomeh, I. Abu Farha,

N. Habash, S. Khalifa, A. Keleg, H. Haddad, I. Zitouni, K. Mrini, and R. Almatham, eds., Singapore (Hybrid), Dec. 2023, Association for Computational Linguistics, pp. 620–624.

[128] N. A. OTHMAN, D. S. ELZANFALY, AND M. M. M. ELHAWARY, *Arabic fake news detection using deep learning*, IEEE Access, (2024).

[129] S. J. PAN AND Q. YANG, *A survey on transfer learning*, IEEE Transactions on Knowledge and Data Engineering, 22 (2010), pp. 1345–1359.

[130] J. PENNINGTON, R. SOCHER, AND C. D. MANNING, *Glove: Global vectors for word representation*, in Conference on Empirical Methods in Natural Language Processing, 2014.

[131] A. RADFORD AND K. NARASIMHAN, *Improving language understanding by generative pre-training*, 2018.

[132] C. RAFFEL, N. SHAZEER, A. ROBERTS, K. LEE, S. NARANG, M. MATENA, Y. ZHOU, W. LI, AND P. J. LIU, *Exploring the limits of transfer learning with a unified text-to-text transformer*, J. Mach. Learn. Res., 21 (2020).

[133] H. RAHAB, A. ZITOUNI, AND M. DJOUDI, *Sana: Sentiment analysis on newspapers comments in algeria*, Journal of King Saud University - Computer and Information Sciences, 33 (2021), pp. 899–907.

[134] S. ROSENTHAL, N. FARRA, AND P. NAKOV, *SemEval-2017 task 4: Sentiment analysis in Twitter*, in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), S. Bethard, M. Carpuat, M. Apidianaki, S. M. Mohammad, D. Cer, and D. Jurgens, eds., Vancouver, Canada, Aug. 2017, Association for Computational Linguistics, pp. 502–518.

[135] A. ROZOVSKAYA, R. SPROAT, AND E. BENMAMOUN, *Challenges in processing colloquial arabic*, in Proceedings of the International Conference on the Challenge of Arabic for NLP/MT, London, UK, 2006, pp. 4–14.

[136] H. SAADANE AND N. HABASH, *A Conventional Orthography for Algerian Arabic*, in Proceedings of the Second Workshop on Arabic Natural Language, Beijing, China, 2015, pp. 69 – 79.

[137] A. SAFAYA, M. ABDULLATIF, AND D. YURET, *Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media*, in Proceedings of the Fourteenth Workshop on Semantic Evaluation, Barcelona (online), Dec 2020, pp. 2054–2059.
Available: `https://www.aclweb.org/anthology/2020.semeval-1.271`.

[138] F. SHAMMARY, Y. CHEN, Z. T. KARDKOVACS, M. ALAM, AND H. AFLI, *Tf-idf or transformers for arabic dialect identification? itflows participation in the nadi 2022 shared task*, in Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP), H. Bouamor, H. Al-Khalifa, K. Darwish, O. Rambow, F. Bougares, A. Abdelali, N. Tomeh, S. Khalifa, and W. Zaghouani, eds., Abu Dhabi, United Arab Emirates (Hybrid), Dec. 2022, Association for Computational Linguistics, pp. 420–424.

[139] A. SHEHATA, M. N. AL-SUQRI, N. E. ELSHAIEKH, F. HAMAD, Y. N. ALHUSAINI, AND A. MAHFOUZ, *Arabfake: A multitask deep learning framework for arabic fake news detection, categorization, and risk prediction*, IEEE Access, (2024).

[140] W. SHISHAH, *Jointbert for detecting arabic fake news*, IEEE Access, 10 (2022), pp. 71951–71960.

[141] S. SHON, A. ALI, Y. SAMIH, H. MUBARAK, AND J. GLASS, *Adi17: A fine-grained arabic dialect identification dataset*, in ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 8244–8248.

[142] A. SMITH, *Turkey earthquake: Four ways tech is being used to help victims*. `https://www.context.news/big-tech/turkey-earthquake-four-ways-tech-is-being-used-to-help` 2023.
Online; 10 Feb. 2023.

[143] B. SONG, C. PAN, S. WANG, AND Z. LUO, *Deepblueai at wanlp-eacl2021 task 2: A deep ensemble-based method for sarcasm and sentiment detection in arabic*, in Proceedings of the Sixth Arabic Natural Language Processing Workshop, N. Habash, H. Bouamor, H. Hajj, W. Magdy, W. Zaghouani, F. Bougares, N. Tomeh, I. Abu Farha, and S. Touileb, eds., Kyiv, Ukraine (Virtual), apr 2021, Association for Computational Linguistics, pp. 390–394.

[144] R. SOUNDARAPANDIAN, *Natural Language Processing in E-Commerce-Enhancing Customer Experience*, Academic Guru Publishing House, 2024.

[145] R. SUWAILEH, M. IMRAN, AND T. ELSAYED, *Idrisi-ra: The first arabic location mention recognition dataset of disaster tweets*, in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 16298–16317.

[146] B. TALAFHA, M. ALI, M. E. ZA'ATER, H. SEELAWI, I. TUFFAHA, M. SAMIR, W. FARHAN, AND H. T. AL-NATSHEH, *Multi-dialect arabic bert for country-level dialect identification*, in Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, 2020, pp. 15–24.

[147] B. TALAFHA, M. ALI, M. ZA'TER, H. SEELAWI, I. TUFFAHA, M. SAMIR, W. FARHAN, AND H. AL-NATSHEH, *Multi-dialect arabic bert for country-level dialect identification*, 2020.

[148] B. TALAFHA, W. FARHAN, A. ALTAKROURI, AND H. AL-NATSHEH, *Mawdoo3 AI at MADAR shared task: Arabic tweet dialect identification*, in Proceedings of the Fourth Arabic Natural Language Processing Workshop, W. El-Hajj, L. H. Belguith, F. Bougares, W. Magdy, I. Zitouni, N. Tomeh, M. El-Haj, and W. Zaghouani, eds., Florence, Italy, aug 2019, Association for Computational Linguistics, pp. 239–243.

[149] R. L. TRASK, *Language: the basics*, Routledge, 2003.

[150] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, L. KAISER, AND I. POLOSUKHIN, *Attention is all you need*, in Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Red Hook, NY, USA, 2017, Curran Associates Inc., p. 60006010.

[151] Q. WEN, *A study on strategies to achieve cross-linguistic communication in the context of internationalisation*, in Proceedings of the 2nd International Conference on Education, Language and Art (ICELA 2022), Atlantis Press, 2023, pp. 788–795.

[152] O. ZAIDAN AND C. CALLISON-BURCH, *Arabic dialect identification*, Computational Linguistics, 40 (2014), pp. 171–202.

[153] R. ZBIB, E. MALCHIODI, J. DEVLIN, D. STALLARD, S. MATSOUKAS, R. SCHWARTZ, J. MAKHOUL, O. F. ZAIDAN, AND C. CALLISON-BURCH, *Machine translation of Arabic dialects*, in Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, E. Fosler-Lussier, E. Riloff, and S. Bangalore, eds., Montréal, Canada, jun 2012, Association for Computational Linguistics, pp. 49–59.

[154] Y. ZHU, R. KIROS, R. ZEMEL, R. SALAKHUTDINOV, R. URTASUN, A. TORRALBA, AND S. FIDLER, *Aligning books and movies: Towards story-like visual explanations by watching movies and reading books*, in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 19–27.