University of Zawia
Faculty of Science
Department of Computer Science

Named Entity Recognition for Libyan Dialect using a Machine Learning Algorithm

Student Name: Lila Saad Arebi

Name of Supervisor: Dr. Ramadan Alsyed Alfared
(Associate Professor)

Thesis Was Submitted in Partial Fulfillment of the Requirements for The Degree of Master in Computer Science.
2026

Declaration

I, Lila Saad Mohammed Arebi, hereby declare that the work contained in this thesis/dissertation, unless otherwise indicated, is my own original work and has not been previously submitted to fulfill the requirements of an award at this university or any other institution of higher education or research. I also hereby waive the copyright to this thesis/dissertation in favor of the University of Zawiya.

Student Name: Lila Saad Mohammed Arebi

ACKNOWLEDGMENT

Praise be to Allah, who has blessed me with the strength, patience, and determination to complete this research.

I dedicate this effort to the pure souls of my departed parents, who instilled in me a love for knowledge and perseverance. I ask Allah to envelop them in His vast mercy and to place this work in the balance of their good deeds.

I extend my deepest gratitude to my supervisor, Dr. Ramadan Al-Fared, for his steadfast support and valuable guidance throughout this research journey. He has been a great help and a kind professor—may Allah reward him with the best reward.

I also express my profound thanks to my husband, my children, and my brothers and sisters for their constant encouragement.

My thanks also go to the management of the Computer Science Department and to everyone who contributed to the completion of this work, even with a prayer or a kind word. To all of you, I offer my sincere appreciation and gratitude.

And our final prayer is that all praise belongs to Allah, Lord of all the worlds.

Abstract

This study evaluates the performance of two deep learning models, AraBERT and BiLSTM, on the task of Named Entity Recognition (NER) for texts written in the Libyan Arabic dialect, a low-resource variant of Arabic with limited annotated corpora. To address this gap, a manually labeled dataset was constructed using the BIO tagging scheme, covering four entity types: Persons (PER), Organizations (ORG), Locations (LOC), and Outside tokens (O), derived from a Twitter corpus consisting of approximately 38,726 tokens. Both models were trained and evaluated using accuracy, precision, recall, and F1-score.

The findings demonstrate strong performance from both models despite the linguistic complexity of dialectal Arabic. The BiLSTM model achieved an overall accuracy of 94% and **a** macro F1-score of 0.82 (PER: 0.85, LOC: 0.83, ORG: 0.75, O: 0.97), reflecting its robustness in handling dialectal variation and irregular morphology. Conversely, AraBERT showed superior recall for Person entities (F1 = 0.88), benefiting from its pre-training on Modern Standard Arabic, while both models exhibited comparable difficulty in detecting Organization entities (ORG).

These results highlight the pressing need to expand Libyan-dialect linguistic resources and point to promising research directions, including dialect-specific pre-training, integrating CRF layers for improved sequence labeling, and examining code-mixed dialect–MSA text. The study contributes to advancing NLP research on Arabic dialects and enhancing NER systems for low-resource languages

Keywords: Arabic Named Entity Recognition (NER); Deep Learning; AraBERT; BiLSTM; Libyan Dialect; Text Mining; Social Media Data; BIO Tagging Scheme.

الملخص

يهدف هذا البحث إلى تقييم أداء نموذجي التعلم العميق AraBERTو BiLSTM في مهمة التعرف على الأسماء (NER) في نصوص اللهجة الليبية، وهي من اللهجات العربية التي تعاني ندرةً واضحة في الموارد اللغوية المُلصقة. ولتجاوز هذا القصور، تم بناء مجموعة بيانات مُعنونة يدويًا بالاعتماد على مخطط BIO لأربع فئات من الكيانات: الأشخاص(PER) ، المنظمات(ORG) ، المواقع(LOC) ، والكلمات خارج الكيانات(O) ، وذلك بالاعتماد على عيّنة من تغريدات تويتر بلغ حجمها نحو 38,726توكن .وتم تدريب النموذجين وتقييمهما باستخدام مقاييس الدقّة، والدقّة الإيجابية(Precision) ، والاستدعاء(Recall) ، ودرجة.F1

أظهرت النتائج أن كلا النموذجين يقدّم أداءً قويًا رغم التحديات البنيوية والصرفية التي تطرحها النصوص اللهجية. فقد حقق نموذج BiLSTM دقّة عامة بلغت 94% ودرجة F1 Macro = 0.82 (PER: 0.85، LOC: 0.83، ORG: 0.75، O: 0.97)، مظهراً قدرة عالية على استيعاب التنوع اللهجي والأنماط الصرفية غير المنتظمة. في المقابل، أظهر نموذج AraBERT تفوقًا في استدعاء الكيانات من نوع الأشخاص(F1 = 0.88) ، مستفيدًا من تدريبه المسبق على نصوص العربية الفصحى، بينما واجه النموذجان صعوبات مشتركة في التعرف على كيانات المنظمات.(ORG)

وتُبرز هذه النتائج الحاجة الملحّة إلى إثراء الموارد الرقمية المخصصة للهجة الليبية، كما تشير إلى مسارات بحث مستقبلية واعدة، تشمل: تطوير نماذج مُدرّبة مسبقًا خصيصًا للهجات، وإضافة طبقات CRF لتعزيز دقة التصنيف المتسلسلي، ودراسة النصوص المختلطة بين اللهجة والفصحى. وتُسهم هذه الدراسة في تعزيز الجهود البحثية ضمن مجال معالجة اللغات الطبيعية للهجات العربية، وفي دعم تطوير أنظمة NER للغات منخفضة الموارد.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Term (English)
ACC	Accuracy
AI	Artificial Intelligence
API	Application Programming Interface
AraBERT	Arabic Bidirectional Encoder Representations from Transformers
BERT	Bidirectional Encoder Representations from Transformers
BIO	Beginning, Inside, Outside
BiLSTM	Bidirectional Long Short-Term Memory
B-tag	Beginning tag
CA	Classical Arabic
CNNs	Convolutional Neural Networks
CRF	Conditional Random Fields
CSV	Comma Separated Values
DL	Deep Learning
F1-Score	F1-Score
FN	False Negatives
FP	False Positives
GPT	Generative Pre-trained Transformer
GPUs	Graphics Processing Units
GRU	Gated Recurrent Units
HMM	Hidden Markov Models
I-tag	Inside tag
LLMs	Large Language Models
LOC	Location (in Entity Classification Context) / Locations
LR	Learning Rate
LSTM	Long Short-Term Memory
ML	Machine Learning
MLM	Masked Language Model
MSA	Modern Standard Arabic
NER	Named Entity Recognition
NLP	Natural Language Processing
NSP	Next Sentence Prediction
O-tag	Outside tag
ORG	Organizations
PER	Persons
POS	Part-of-Speech
PREC	Precision
REC	Recall
RNN	Recurrent Neural Networks
SVM	Support Vector Machines
TN	True Negatives
TP	True Positives

Chapter One:

Introduction

1.1 Background

Named Entity Recognition (NER) is an essential component of Natural Language Processing (NLP), which helps to automatically recognize and categorize objects in text (e.g., names of persons, companies, places, and dates) [1]. This has been the foundation of many NLP programs, such as information retrieval, question answering, and machine translation [2].

Nevertheless, the creation of efficient NER systems is even more complicated in the situation when it is implemented in the dialectal forms of a language, which are still in their underdeveloped state, compared with Modern Standard Arabic (MSA). In spoken forms like the Libyan, phonological, lexical, and syntactic aspects are significantly different from MSA, as pointed out by Owens [3]. They further complicate the existing NLP tools, which were mainly trained and optimized on MSA, hence deteriorate when they are used in dialects [4].

The weaknesses within the NLP studies related to Libyan dialect (and Arabic dialects, in general) are the lack of annotated corpora and language materials [5]. Thus, the majority of off-the-shelf NER systems that have been trained on MSA or more well-documented dialects (e.g., Egyptian or Levantine Arabic) have poorer performance when applied to Libyan Arabic due to linguistic mismatch. This is the gap that highlights the importance of specialized datasets and models that are specific to the Libyan dialect.

New developments in the area of deep learning provide good answers to these challenges. Recurrent neural networks based on LSTM and Transformer-based models like BERT have proven to be the state of the art with regard to modeling context structures and dependency structures in natural language [6]. Such models are able to acquire dialect-specific patterns well, as long as they are provided with large and representative annotated sets [7].

The given research aims to address the lack of named entity recognition (NER) resources that are applicable in the case of Libyan Arabic dialect by creating a machine-learning-based NER engine that is specifically trained to deal with this language variant.

To achieve such a goal, one will need to build a high-quality corpus that represents the distinguishing variables of the dialect, fine-tune the preprocessing pipelines, and select the machine-learning models that can accept the dialect-specific variations yet produce the same performance metrics as the established standards.

More than just its direct effect on Libyan natural language processing, the work presented in the paper builds upon larger projects that the research seeks to promote the concept of linguistic inclusivity in computational resources. Enriching and improving dialectal and under-resourced language tools not only contributes to cultural heritage preservation however, it also extends the application of NLP technologies to the heterogeneous linguistic population.

1.2 Problem Statement

The emergence of digital textual materials has increased the need for efficient natural language processing (NLP) systems that can process and organize information effectively. An essential part of such work is named entity recognition (NER), which enables the recognition and structuring of named entities. However, the currently implemented NER systems tend to have suboptimal performance with dialectal Arabic, partly owing to their strong dependency on resources that have been mostly created for Modern Standard Arabic (MSA).

The situation is further complicated by the Libyan dialect, which exhibits both phonological and syntactic divergence from MSA and suffers from a marked scarcity of annotated corpora. These challenges hinder the development of NLP tools tailored to Libyan Arabic. Consequently, there is an urgent need for NER models and resources capable of effectively addressing the linguistic characteristics of the Libyan dialect and supporting the processing of its online textual content.

1.3 Research Objectives

1. Building and classify an annotated corpus for the Libyan dialect, comprising named entities distributed across specific categories such as people, places, and organizations, to serve as a training and reference base for evaluating deep learning models.

2. Design and implement a named entity recognition (NER) model based on the Bi-LSTM bidirectional short-term memory architecture, and evaluate its efficiency in extracting entities from unfocalized Libyan texts.

3. Explore the performance of pre-trained large language models based on transformer-based architectures, such as AraBERT, and fine-tune them to perform the named entity recognition task in the Libyan dialect.

4. Conduct a comprehensive comparative analysis of the performance of the two implemented models (Bi-LSTM and the transformer-based model) using standard evaluation metrics (such as accuracy, recall, and F1 score), and identify the optimal model that achieves the highest performance levels in the context of the Libyan dialect.

5. Identifying the specific linguistic and computational challenges facing the recognition of named entities in the Libyan dialect (such as polymorphism, derivation, non-standard notation, and lack of data resources), and providing clear recommendations for developing future research efforts in addressing Arabic dialects.

1.4 Research Questions

1. What linguistic properties of the Libyan dialect have the strongest impact when it comes to the performance of named entity recognition systems?
2. Which are the most effective machine learning methods to use in improving named entity recognition in the Libyan Arabic dialect?
3. What is the comparison of different machine learning models with regard to NER performance on Libyan Arabic, which has been analyzed based on precision, recall, and F1-score?
4. How does the quality and availability of annotated data influence the NER system outputs of dialectal Arabic?

1.5 Thesis Structure

Chapter 1: Introduction

The introduction chapter provides a complete background of the study, outlining the contextual background and theoretical support of the study. It states the research problem, outlines the objectives, states the research questions, and elaborates on the reasons why the creation of a Named Entity Recognition system was necessary in the specifics of the Libyan dialect.

Chapter 2: Background to Artificial Intelligence and Deep Learning

This part presents a logical explanation of the theoretical background of Artificial Intelligence, Machine Learning, and Deep Learning, which play a critical role in the application of Natural Language Processing. It traces the history of the shift of traditional paradigms of computations toward advanced frameworks like BiLSTM and AraBERT, which is the backbone of modern NER systems.

Chapter 3: Literature Review

It is the critical review of the available literature on NER and machine learning approaches, specifically focusing on Arabic and its dialects. It finds the methodological and linguistic issues due to the dialectal diversity and enumerates the existing solutions, such as the traditional methods, deep learning models, transformer-based architectures, and cross-dialectal transfer learning literature. The review highlights the lack of annotated materials on dialectal Arabic and that of Libyan.

Chapter 4: The Natural Language Processing and the Libyan Dialect.

The given chapter is focused on the NLP related to the Libyan dialect, which outlines the linguistic features, morphological complexity, and orthographic variants that impact NER efficiency. It also highlights the need to design dialect-oriented models and preprocess strategies to effectively deal with the distinctive features of language.

Chapter 5: Methodology

The methodology framework is presented in a comprehensive manner and includes research design, data collection processes, annotation procedures, preprocessing strategies, model selection criteria, and evaluation process. It clearly explains how the algorithms of machine learning and deep learning are modified to meet the demands of the Libyan dialect.

Chapter 6: Results and Discussion.

In this chapter, the chapter introduces the experimental design, quantitative results of performance of the proposed NER model, and comparative analysis with other methods. The testing utilizes uniform measures in the evaluation process and evaluates the effectiveness of the model in reflecting dialectal linguistic systems.

Chapter 7: Conclusion and Future Work.

The main research conclusions are outlined, the disadvantages are discussed, and the future research purposes are discussed. Several issues, including the promotion of dialectal NER, the expansion of the resources of underrepresented languages, and the investigation of promoting the Arabic NLP applications based on the advanced machine learning and transformer-based models, are prioritized.

Chapter Two:
Background to Artificial Intelligence and Deep Learning

2.1 Introduction

The broad definition of Artificial Intelligence (AI) is the science and engineering of creating intelligent machines [8]. On its part, AI, as a specific sub-discipline of the field of computer science, aims at imitating human-like cognition in a computer system, which can play the role of a human by solving the problems traditionally requiring the human mind to be resolved [8]. The modern AI systems have been designed to perform reasoning, problem-solving, decision-making, and pattern-recognition in a wide range of areas, such as healthcare, finance, autonomous systems, and natural language understanding [9].

Machine Learning (ML) within the domain of AI focuses on providing computer systems with the ability to learn and be driven by sensory information and practice instead of solely being controlled by direct instructions that are coded within the programming at the time they are created [10]. The use of ML methodologies gives machines the ability to internalize new information and distinguish pattern, as well as make predictions or decisions on their own. The ability to extrapolate from exemplars underlies most AI Applications to date, including image recognition, speech processing, predictive analytics, and autonomous navigation [11].

The advent of ML created a paradigm shift in the research in AI [10]. Before its introduction, intelligent system development required tedious and rule-based programming, which required the developer to encode knowledge manually in every conceivable situation [10]. Conversely, ML enables systems to build predictive models with direct data, which helps in achieving scalability, flexibility, and performance excellence in multifaceted, real-world activities[8].

2.2 Artificial Intelligence (AI)

AI has become one of the most powerful Domains of modern computer science and should be subject to serious academic scrutiny [8]. Its main aim is to inculcate systems that are able to conform to human-like cognition in a variety of dimensions, which include decision-making, reasoning, perception, language understanding, and tactics [8]. The hypothetical support structure of AI was built during the 1950s, with such key figures as Alan Turing, who originally proposed the idea of a thinking machine, and John McCarthy, who officially named the field of study Artificial Intelligence [8][9].

AI has evolved into more practical applications that dominate daily lives over a series of decades, developing from a theory to a practical application [8]. AI applications in the modern day cover natural language processing (NLP), computer vision, robotics, expert systems, and mass optimization in industrial infrastructures [12]. In addition, AI has a leading role to play in the management and processing of big data, which allows effective analysis and automatic decision-making in a range of industries.

2.3 Machine Learning (ML)

Machine learning is one of the core foundations of artificial intelligence, in which algorithms are created to allow machines to learn directly from data [10]. The ML algorithms do not require explicit programming of all possible scenarios; instead, of identify a pattern, create predictive models, and optimize an improvement based on experience [12].

Machine learning can be traced back to the early 1940s when researchers started to focus on artificial neural networks, which were later continued in the 1950s when a perceptron was developed [12]. In several decades, several other algorithms, including decision trees, k-nearest neighbors, support vector machines, and probabilistic models, came up, and they provided the building blocks of modern AI systems [9].

2.3.1 Types of Machine Learning

Machine Learning covers a diversity of methods depending on the way models derive knowledge out of information:

1. Supervised Learning: with labeled data, it comes to an inference about a correspondence between inputs and the desired outputs. Regular uses are in classification, regression, and predictive modeling [13].
2. Unsupervised Learning: does analysis on unlabeled information in order to expose concealed tendencies or clusters. They can be used in anomaly detection, market segmentation, and dimensionality reduction [10].
3. Semi-Supervised Learning: this combines both labeled and unlabeled data and provides a compromise between performance improvement and less labeling work [14].

4. Reinforcement Learning: involves the use of environmental feedback that is used to optimize sequential decisions. It is being extensively used in game playing, autonomous navigation and robotic control [16].
5. Self-Learning: It is used in systems that discover knowledge independently of the system being guided without the help of any external system, which is commonly used in adaptive systems [16].

2.3.2 Other Learning Approaches

Besides the classical paradigms, sophisticated machine learning approaches have been implemented to solve multifaceted problems and optimize the efficiency of the model:

- Transfer Learning: this technique means the use of knowledge acquired to enhance other related task performance [12].
- Adversarial Learning: In order to improve model resistance, an adversarial or perturbed example is presented to the model [17].
- Feature/ Representation Learning: Leader in automated derivation of useful data representations, thus decreasing the use of hand-crafted feature engineering [18]
- Sparse Dictionary Learning: Signal representations that replicate their original using sparse, interpretable bases, a method used in signal processing and computer vision [19].

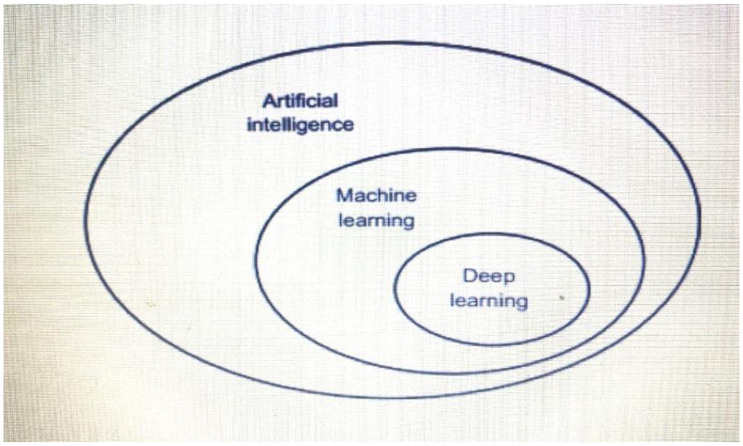

Figure 2.1 AI, ML, and DL

The visual representation shows the hierarchy of the relationship between Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL).

2.4 Deep Learning (DL)

Deep Learning (DL) is a subdivision of machine learning that uses deep neural networks to autonomously acquire hierarchical data representations. DL architectures are specifically very efficient in the context of capturing non-linear interdependencies in vast datasets. Fast hardware progress- more specifically, the graphic processing units (GPU) - and increasing access to large, annotated corpora have only contributed to further increasing the power of DL [20].

Deep neural networks consist of sequential directions of associational connected neurons:

2.4.1 Deep Learning Process

- Forward Pass: Calculates weighted averages of activation at the input ends, and then the nonlinear activation functions are used [20].
- Backward Pass: It uses back-propagation to calculate the gradients of a loss function with respect to each weight, to do the iterative parameter update [18].

In contrast to the traditional machine-learning models, the DL device eliminates the necessity of the manual construction of features; the networks naturally learn the high-level representations on the raw input information. This end-to-end learning model makes DL remarkably useful to various tasks, such as image recognition and processing speech, autonomous driving, and natural-language understanding [20].

2.4.2 Deep Learning in Named Entity Recognition (NER)

Deep learning has transformed named entity recognition (NER) through free-learning mechanisms that acquire and extract contextual and semantic features of an annotated corpus, reducing the need to oversee a set of manually developed sets of rules [21]. Co Key DL architectures used in NER include:

- Convolutional Neural Networks (CNNs): CNNs are networks that absorb local contextual lexical information in lexical sequence series [20].
- Recurrent Neural Networks (RNNs): RNNs are effective since they handle sequential information whilst maintaining temporal memory [21].
- Long Short-term Memory (LSTM) and Bidirectional LSTM (BiLSTM): Here, long-range dependencies in both directions are displayed, both forward and backward, and are

commonly combined with Conditional Random Fields (CRFs) in order to yield accurate sequence labelling [22].

2.5 Hierarchical Relationship Between AI, ML, and DL

One way of thinking of the connection between AI, ML, and DL is a hierarchy:

- **AI**: It is all intelligent computing.
- **ML**: Gives the processes of learning and adaptation in the AI systems.
- **DL**: It is a more sophisticated form of ML and it will exploit deep neural networks to represent intricate patterns and relationships.

The hierarchical structure above shows how deep learning simplifies the traditional machine-learning methods to much more complicated, unstructured, high-dimensional data, thereby augmenting the intelligence and autonomy of computational systems.

2.6 Summary

Overall, the chapter has provided a theoretical framework for understanding the evolution and stratification of intelligence in computational systems. It has started with a definition of artificial intelligence and its ability to simulate human cognitive processes, the principles and paradigms of machine learning, and focused on the input of the machine learning principles data-driven learning to provide intelligent flexibility, and concluded by introducing deep learning as an advanced branch of machine learning and able to acquire hierarchical representations, highlighting its disruptive impact on modern data-based applications.

Chapter Three
Literature Review

3.1 Introduction

Over the last few years, Arabic Named Entity Recognition has seen significant development. Initially, early methods used rule-based systems and features that had to be manually engineered with a lot of linguistic knowledge, often failing to account for context variation. Since the introduction of machine learning (and more recently deep learning) methodologies have drifted towards architectures that are able to learn semantic representations and contextual representations independently [23]. However, there are unique difficulties with Arabic:

a complex system of morphology, mixed syntax, and many dialects. These characteristics are detrimental towards creating strong NER systems, particularly those of under-resourced dialects like the Libyan Arabic [24]. The chapter thereby gives an overview of the landmark works in the Arabic NER, clarifies the current research methods orientation, outlines the current challenges, and highlights gaps that catalyze the development of dialect-particular systems [25].

3.2 Previous Studies: Arabic Named Entity Recognition

The field of Arabic Named Entity Recognition (NER) has seen significant progress in the past ten years, ranging from the typical feature-driven approach to the employment of deep learning and transformer models, alongside dealing with the dialectal and social media aspects [22].

3.2.1 Traditional Approaches:

Early Arabic Named Entity Recognition models were mostly rule-based, incorporating machine learning techniques alongside these rules [26]. The features included were parts of speech, lexical features, suffix features, and definite articles. Later in 2018, Aldali came up with a holistic model that combines these features with Support Vector Machines(SVM), Maximum Entropy(ME), and Conditional Random Fields techniques(CRF). Using a voting ensemble strategy, the proposed model reached an F1 value of 94.21% on recognising organizational entities [27]. It can be concluded from the above observation that models with a wide range of features can be used for efficient recognition of a few categories with high accuracy. however, with very high computational complexity due to the design of features [28].

3.2.2 Deep Learning Approaches:

With the rise of deep learning, it has now become possible to automatically extract complex patterns in data, which means that feature engineering is no longer needed. Elbazi and Laachfoubi [29] came up with a BiLSTM-CRF model that included both pre-trained word embedding and character-level representations to reach an F1-score of 90.6 per cent on the ANERcorp dataset [29]. LSTM was a two-way device that was used to retrieve the context of the previous and following tokens, and the CRF layer was used to enhance the quality of sequence labeling. On the same note, Khan et al. [30] used deep-learning methods on named-entity recognition and entity linking on Arabic tweets and trained their model with geolocation data contained in Google Maps [30]. Their system achieved 40-percent accuracy at the regional scale, 48-percent at the city scale and 63-percent accuracy on points of interest, thus highlighting the problems of informal social-media content.

3.2.3 Hybrid Approaches:

Knowing the expressive power of pre-trained transformer embedding's, hybrid models are built on top of that to exploit the sequential power of recurrent neural networks to add to the contextual understanding. Alsaaran and Alrabiah [31] combined AarBERT embedding's with a Bidirectional Gated Recurrent Unit (BGRU) to achieve F1 -scores of 92.28 -percent on the ANERcorp corpus and 90.68 -percent on a merged ANERcorp -AQMAR dataset. Their results highlight the concrete advantages of combining contextual representations of words with deep sequence learning architectures in order to tackle the complex syntactic and lexical challenges that Arabic faces [31].

3.2.4 Transformer-Based Approaches:

Transformer-based models, especially BERT, have triggered transformative changes in Arabic Named Entity Recognition. A study by Al-Qurishi and Souissi [32] used AraBERT with a Conditional Random Field (CRF) decoding layer, which obtained macro-averaged F1-scores of 89.6, ANERcorp, and 88.5, AQMAR. Continuing this trend, Jarrar et al. published the so-called Wojood corpus in 2022, which is full of 21 entity types with 22.5 percent nested annotations [33]. Their AraBERT-based multi-task model achieved a micro F1-score of 88.4% and thus provided a strong benchmark of nested Arabic NER and also demonstrated the transformative quality of highly annotated datasets on model results.

3.2.5 NER in Social Media and Colloquial Arabic:

The spread of the informal and dialectical textual materials brings with it a range of other complications. A rule-based system by Alfared and Alhammi (2018) [25], adapted to the Libyan Arabic language, was able to find 8,583 named entities, consisting of 71.59% of persons, 26.13% of places, and 2.27% of companies. Developing the theme of rule-based approaches, Khan et al. [30] showed that entity recognition on Arabic tweets can be improved further with the convergence of deep learning methods and metadata of geolocation, regardless of the noisiness and unstructured characteristics of social media information [30].

3.2.6 Arabic Dialect Classification:

Dialect identification is one of the critical conditions of successful NER in user-generated material. The 2021 usage of AraBERT and AraELECTRA by Wadhawan, alongside the Farasa segmentation, gave the macro F1 -score of 0.235 in the NADI shared task framework [34]. In line with this, Talafha et al. [35] combined several versions of BERT in dialect classification, achieving a micro F1 -score of 26.78 percent at the country level [35]. Lastly, Younes et al. [24] worked on Romanized Tunisian Arabic and used BiLSTM-CRF architecture that uses FastText embedding's and had an accuracy of 98.65% [24]. Together, they provide insights into the unparalleled significance of dialect-conscious NER systems and outline new directions in the context of the research of this subtle area.

3.2.7 Maghrebi Dialects and Cross-Dialectal Studies:

Studies of the Maghrebi dialects highlight the incessant issues that emerge due to the lack of resources and syntax diversity. Harrat and colleagues [36] highlighted that corpora and machine-translation tools are not standardised with Algerian, Moroccan and Tunisian varieties [36]. More recently, Chrif and colleagues [38] have used classical machine-learning tools, Random Forest, Logistic Regression, and Naive Bayes, to divide Mauritanian dialect IC posts on Facebook [38]. Their Random Forest model achieved a performance of 96.37 ± 0.01, indicating high classification accuracy and strong result stability. This demonstrates the significant potential of existing algorithms when trained on sufficiently large and well-annotated datasets

3.2.8 Sentiment Analysis in Arabic:

Methods derived from sentiment analysis are also used to drive name-entity recognition studies, especially in short texts and data-sparse settings. Al-Yousef [23] used Word2Vec embedding's along with a BiLSTM neural network and transfer learning to work around morphological complexity and finally outperform traditional models on the Arabic Sentiment Tweets Dataset [23].

In a previous study by Aljarmi [37], the research addressed sentiment analysis on Twitter in the Libyan dialect, aiming to fill a gap in resources specific to this dialect. The study relied on applying classical machine learning algorithms such as Decision Tree, SVM, and Naïve Bayes to a dataset consisting of 3,000 tweets. The results showed that the Decision Tree algorithm achieved the best accuracy, at 83%, followed by SVM at 81%, while Naïve Bayes achieved 80% [37].

3.2.9 Cross-Dialectal Transfer Learning:

Dialectal NER can be done through transfer learning with adequate local resources, particularly, the resources of Modern Standard Arabic (MSA). Elkhbir et al. [39] studied the impact of zero-shot transfer learning and showed that pre-trained language models obtain relatively good performance on dialects of the Egyptian language, the Moroccan and the Syrian language without the need to be further annotated. This highlights that it is promising to generalise MSA-trained models to low-resource dialects [38].

3.3 NER in Dialects and Social Media

NER systems of dialectal Arabic also face further challenges due to informal orthography, code -switching and Romanized script. Studies have covered various dialects, such as Egyptian Arabic, Maghrebi Arabic, Levantine Arabic, and Gulf Arabic, and have used deep learning models that have been trained with social media text to identify objects in short, noisy and unstructured texts. In addition, geolocation and entity linkage has been incorporated to give the location-based entities a background [30].

Dialect classification- This is essential to successful NER, because dialect differentiation allows one to choose the right pre-processing and model-adaptation techniques. Models based on transformers and segmentation methods have shown the ability to categories dialects, although the level of performance is limited when working in low resource conditions like Libyan Arabic [34].

3.4 Challenges in Arabic NER

Despite progress, several persistent challenges affect Arabic NER systems:

1. **Morphological Complexity**: The lexical items in the Arabic language have a rich network of inflexion and derivation morphology, which puts significant ambiguity in the task of entity recognition [26].
2. **Orthographic Deviations**: Spelling differences and unsystematic word structure make the systematic process of determining entities of the Arabic corpus even more complicated [35].
3. **Dialectal differences**: The difference between Regional varieties and Modern Standard Arabic in lexical selection and syntactic structure as well as phonological display, restricts the effectiveness of cross-dialectal models [34].
4. **Lack of resources**: Most annotated corpora are biased towards MSA; this makes dialectal registers, in particular, the Libyan Arabic, significantly underserved [38].
5. **Nested and Discontinuous Entities**: Nested or discontinuous entity spans are highly common in the Arabic literature, a phenomenon that requires complex sequence-modeling methods [33].

These challenges highlight the need for models that can generalize across dialects and handle complex linguistic structures.

3.5 Research Gaps

Analysis of the literature reveals specific gaps in Arabic NER research:

- There is an insufficient number of annotated corpora of Libyan dialect that would unquestionably limit the creation of strong models [39].

- The lack of research on hybrid architectures and transformer-based formulas that are specifically tailored in terms of under-resourced dialectal situation [38].
- An inadequate investigation of cross-dialectal transfer-learning procedures.
- poor methodological evaluation of the dialectical recognition of entities by nesting and discontinuity [33].

The urgent need to fill these gaps is the way to make the Arabic NER systems more comprehensive and resilient, particularly in dialectic applications.

3.6 Need for Libyan Dialect NER

Developing a dedicated NER system for Libyan Arabic is crucial. Such a system would:

- Extract named entities in corpora in the Libyan Arabic.
- Mediate between extant resources of MSA to dialectal settings.
- Support downstream uses of nonversation, information search and social monitoring.
- preserve the linguistic and cultural heritage contained in the dialect-based discourse.

These issues enhance the importance that dialect-generalizing models could be developed that could easily deal with complex linguistics.

3.7 Summary

The scientific literature suggests the presence of a significant breakthrough in Arabic NER based on deep-learning and transformer-based systems. Albeit MSA-based systems do achieve rather impressive productivity, the acknowledgment of dialectal units continues to be hampered by a lack of data and high heterogeneity in languages. The dialect of Libyan Arabic in a specific case is severely under-represented thus providing a salient research vector. To overcome this deficiency, the development of specialized datasets, the application of special attention to models adaptation, and strict evaluation procedures will be required, and finally, strong, context-aware systems of NER will be developed specific to under-resourced dialects.

Chapter Four:
Natural Language Processing (NLP) and Named Entity Recognition (NER)

4.1 Introduction

Natural Language Processing (NLP) is an important subfield of Artificial Intelligence that works towards endowing the machines with the capability to understand, read, write, and converse with human language sensibly and wittingly. Being at the intersection of computer science, linguistics and machine learning, NLP provides both theoretical framework and practical techniques used to build systems that have the ability to consume and process rich corpora of textual and spoken information, allowing the integration of human interaction with computational interpretation [40].

The rise of digital means of communication such as email messages, updates on social networking platforms, product reviews posted online, and business documents alongside the mass usage of voice-driven interfaces such as virtual assistants or voice-based command systems has significantly increased the call to have more robust and sensitive NLP solutions. These systems are attempting to model complex linguistic phenomena, including syntax, semantics and pragmatics, to concisely capture salient insights, automate language-centric functionality and support decision-making [41]

Alongside this, NLP has increasingly earned an irreplaceable role in the sphere of knowledge extraction, information retrieval, and human-computer interaction, allowing it to be used in machine translation, sentiment analysis, named entity recognition (NER), and question-answering systems. In the very context of Arabic language processing, NLP faces unique obstacles, due to the morphological richness of the language as well as its dialectal diversity, as well as due to the contextual ambiguity that involves situation-spread ambiguities. The challenges of dealing with them require advanced computational model which can provide both fine-grained and word-level features at the same time as well as global and contextual features, thus forming the basis of intelligent and domain-specific system [41].

All in all, it is true that NLP can not only simplify access to automated understanding of human language, it can also enable AI systems to converse with users using a more natural, interactive system, which leads to increased accessibility and decision support and cultural preservation. The current chapter therefore outlines the general role of NLP, especially on how Named Entity Recognition is applied in Arabic literature, while also encompassing its theoretical implications as well as recent deep-learning algorithms capable of facilitating successful linguistic interpretation [40].

4.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a major sub-discipline of artificial intelligence (AI) concerned with providing computational systems with the ability to use, interact, understand, and communicate with human language in a way that is semantically and intellectually sound. It sits at the nexus of computer science, linguistics and machine learning thus forming the foundation of intelligent systems in charge of handling or examining large volumes of written or spoken information [40] [41].

The radical expansion of online forms of human communication, which can take textual forms (in the form of emails, social-media reports, documents, etc.) and speech forms (as voice commands and virtual assistants) has triggered a significant increase in the demand of reliable NLP technologies. Such systems attempt to cover the gulf between natural human language and mechanistic knowledge due to the computational modeling of linguistic phenomena [42].

4.2.1 Levels of Natural Language Processing (NLP)

NLP is methodically structured in a hierarchical structure of the range of linguistic levels which focus on a specific aspect of language. Such seamless grasp age of those strata gives machines the ability to precisely perceive natural language in order to process it [41].

1. **Phonological Level:** Deals with the study of speech sounds (phonemes), including aspects like stress, intonation, and pronunciation. It is essential for systems that process spoken language input or output [42].
2. **Morphological Level:** Focuses on analyzing the internal structure of words, including roots, prefixes, and suffixes. It allows systems to identify word functions and derive different word forms [42].
3. **Syntactic Level:** Concerns the grammatical structure of sentences, determining how words are arranged and related. Techniques such as Part-of-Speech (POS) tagging and syntactic parsing are used here [43].

4. **Semantic Level:** Interprets the literal meaning of words and phrases, resolving ambiguities in polysemous words and identifying synonymy, which is fundamental for accurate language understanding [41].
5. **Pragmatic Level:** Goes beyond literal meaning by considering the speaker's intent and the socio-linguistic context. It plays a vital role in human-computer interaction where understanding implied meanings is necessary [43].
6. **Discourse Level:** Analyzes relationships between sentences and larger text structures. This includes coreference resolution and discourse coherence, which are crucial in summarization and dialogue systems [44].

4.2.2 Applications of NLP

NLP techniques are applied in a wide range of real-world systems, each utilizing one or more of the levels mentioned above.

1. Machine Translation: Requires syntactic and semantic analysis to accurately translate text between languages [44].
2. Speech Recognition and Text-to-Speech: Relies on phonological and morphological analysis to convert speech to text or vice versa [42].
3. Smart Search Engines: Use NLP to understand user intent, process queries, and retrieve semantically relevant results [45].
4. Chatbots and Virtual Assistants: Use syntax, semantics, and pragmatics to handle user interactions [44].
5. Sentiment Analysis: Detects sentiment or emotional tone in text, widely used in marketing and social media analysis [42].
6. Named Entity Recognition (NER): Identifies and classifies proper names (persons, locations, organizations) in text. It is essential in knowledge extraction and information retrieval [46].
7. Text Summarization: Relies on discourse-level analysis to produce coherent and concise summaries [43].
8. Spelling and Grammar Correction: Employs morphological and syntactic levels to detect and correct linguistic errors [48].

9. Text Classification: Includes categorizing documents or messages into predefined classes (e.g., spam detection) [49].

4.3 Named Entity Recognition (NER): Concepts and Objectives

Remember the Named Entity Recognition (NER), the fundamental component of NLP, is orchestrated in the sense of discovering and categorizing the textual units referring to the real-life entities, such as individuals, organization, geographical locations, dates, and numerical formulations. The ultimate goal of NER is to process unstructured textual data with the help of identifying salient constituents in a text and listing them into well-defined semantic categories. The result of this transformation is a greater mechanical understanding of language and, more fundamentally, handicrafts a wide range of downstream uses, such as information extraction and question answering to machine translation and text summarization.

Besides its main capabilities, NER significantly increases search relevance and retrieval of information as it enables the systems to search documents in terms of including relevant entities. Furthermore, it provides material contribution to the construction of such semantic structures like knowledge graphs and is interoperable with linked data ecosystems.

4.3.1 Techniques Used in Named Entity Recognition (NER)

The techniques used in NER have evolved significantly, from traditional methods to modern deep learning approaches [50].

1. Rule-based Approaches: These methods rely on linguistic rules, dictionaries, and lexicons to identify entities. They are accurate within specific domains yet they lack flexibility.
2. Statistical Methods: Include models such as Hidden Markov Models (HMM), Support Vector Machines (SVM), and Conditional Random Fields (CRF). They learn patterns from labeled data, allowing better generalization.
3. Deep Learning Techniques: Employ neural networks such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Transformer-based models like BERT and GPT. These models leverage contextual information and have significantly improved accuracy.

4. Hybrid Models: Combine rule-based approaches with statistical or deep learning techniques to enhance performance.

5. Word Representations: Word embedding (Word2Vec, GloVe) and contextualized embedding (ELMo, BERT) convert words into numeric vectors, markedly improving

model performance.

Figure (4.1) illustrates the comprehensive hierarchical taxonomy of Named Entity Recognition methods, which classifies them into three main categories: knowledge-based methods, feature engineering-based methods, and deep learning-based methods. This taxonomy highlights the historical evolution of the field—from manual approaches to classical machine learning models, culminating in the most advanced neural architectures. Furthermore, it positions this research within this broad landscape, as it focuses on leveraging the capabilities of Transformer models and hybrid architectures to address the challenges specific to the Libyan dialect.

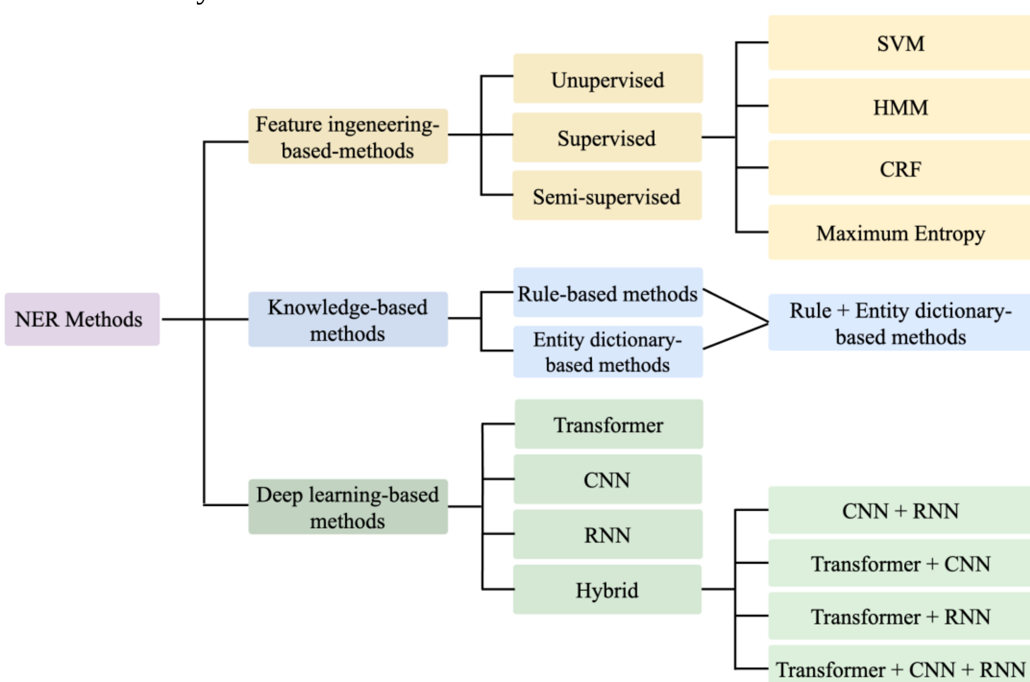

Figure 4.1 Main approaches

4.3.2 Types of Named Entities

Named entities in text can be categorized into different types based on their structure [51]

1. **Nested Named Entities:** Occur when one entity is embedded within another (e.g., "Mokhtar Clinic in Souq Al-Jum'aa, Tripoli"). Identifying them requires hierarchical models.
2. **Continued Named Entities :** Traditional, contiguous entities within the text (e.g., "Saif al-Islam Gaddafi", "Benghazi"). These are handled using sequence labeling models.
3. **Non-Continued Named Entities:** Discontinuous segments of text that refer to the same entity, often interrupted by other content (e.g., descriptions of a medical symptom across a sentence). Recognizing them requires models that capture long-range dependencies, like Transformers.

4.3.3 Practical Applications of NER

NER techniques are widely applied in various real-world scenarios [52]:

- **Information Extraction:** Transforming raw text into structured, machine-readable data.
- **Search Engine Optimization:** Enhancing result relevance by recognizing entities in queries.
- **Question Answering Systems:** Identifying key entities in questions to retrieve precise answers.
- **Biomedical and Clinical Text Processing:** Extracting patient information, drug names, and diseases.
- **Legal and Financial Document Analysis:** Identifying key entities in contracts and reports.
- **Machine Translation:** Preserving named entities across languages.
- **Social Media Analysis:** Tracking events, people, and trending topics.
- **Knowledge Graph Construction:** Building graphs based on entities and their relationships.
- **Development of Intelligent Dialogue Systems:** Improving comprehension of voice assistants.

4.4 Deep Learning Models for Named Entity Recognition (NER)

Deep learning systems have largely revolutionized the performance of Named Entity Recognition (NER) systems, especially when the language is morphologically rich like Arabic. Unlike traditional machine-learning tools that are highly dependent on human feature engineering, the contemporary deep-learning systems are able to automatically generate contextual and semantics representations using massive amounts of corpora. These models not only acquire long-range dependencies, and also understand the immediate situation and make robust generalizations across different areas and types of writing.

Deep learning models for NER are commonly categorized into three major families:

1. **Sequence-based neural models**, such as LSTM and Bidirectional LSTM (BiLSTM).
2. **Convolutional Neural Networks (CNNs)** for character-level and subword feature extraction.
3. **Transformer-based architectures**, such as BERT and its Arabic variants (e.g., AraBERT).

The following sections provide a detailed academic description of the main deep learning models used in Arabic NER.

4.4.1 Bidirectional Long Short-Term Memory (BiLSTM)

Bidirectional Long Short -Term Memory (BiLSTM) networks are the highly developed version of the standard LSTM architecture. They are members of the category of Recurrent Neural Networks (RNNs) which are also engineered to provide models of sequential data. BiLSTM can understand context better than unidirectional LSTMs can since it uses the input sequences going forward and backward, allowing higher comprehension of contexts [53].

A) Architecture of BiLSTM

A BiLSTM network consists of two independent LSTM layers:

- A **forward LSTM layer** that processes the input from left to right.

- A **backward LSTM layer** that processes the input from right to left.

The combined messages of the forward and the backward layer are combined at each time step to create an integrated representation that captures at the same time past and future word dependencies. The two-way contextualizing nature of such coding makes BiLSTM particularly useful in NER tasks, as the model is capable of identifying subtle pattern of relationships on the full sentence scale.

B) Mathematical Formulation of the LSTM Cell

Each LSTM cell contains four gates that regulate information flow:

1. **Forget Gate**

$$f_t = \sigma\left(W_{xf}\, x_t + W_{hf}\, h_{t-1} + b_f\right)$$

Determines which information from the previous cell state should be discarded.

2. **Input Gate**

$$i_t = \sigma(W_{xi}\, x_t + W_{hi}\, h_{t-1} + b_i)$$

Controls which new information should be added.

3. **Candidate Cell State**

$$\tilde{C}_t = \tanh(W_{xc}\, x_t + W_{hc}\, h_{t-1} + b_c)$$

Creates proposed new memory values.

4. **Cell State Update**

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

Combines old and new information.

5. **Output Gate**

$$o_t = \sigma(W_{xo}\, x_t + W_{ho}\, h_{t-1} + b_o)$$

Controls the visible output of the cell.

6. **Hidden State**

$$h_t = o_t \odot \tanh(C_t)$$

Where:

- $\sigma(\cdot)$ is the sigmoid function.
- $\odot$ denotes element-wise multiplication.
- W and b are trainable weights and biases
- x_t is the input at time step t.
- h_{t-1} is the previous hidden state.
- C_{t-1} is the previous memory cell state [54].

C) Illustration

To further clarify the architecture and information flow within the LSTM model, Figure 4.2 illustrates the structure of a bidirectional Long Short-Term Memory (BiLSTM) network. Unlike the standard unidirectional LSTM, the BiLSTM processes the input sequence in both forward and backward directions, allowing the model to capture contextual information from past and future time steps simultaneously. This dual processing enhances the model's ability to learn long-range dependencies, which is particularly beneficial for sequence-labeling tasks such as named entity recognition.

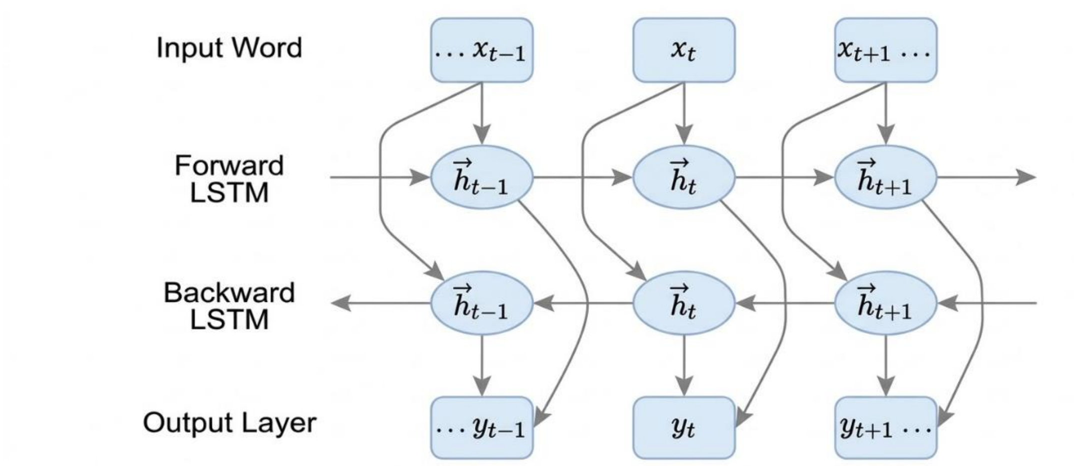

Figure 4.2 A bidirectional LSTM [55]

Table 4.1: BiLSTM Advantages and Disadvantages

Advantages	Disadvantages
Comprehensive Context Understanding: Processes data in both directions, capturing dependencies from past and future.	Increased Computational Complexity: Requires two separate LSTM layers, increasing weights and parameters.
Better Predictive Performance: Often higher accuracy than unidirectional LSTM.	Longer Training Time: Dual processing and more parameters require significantly more resources.
Mitigation of Vanishing Gradient: Uses gating mechanisms to preserve long-term dependencies.	Unsuitable for Real-Time Processing: Must wait for the complete sequence before backward layer operation.
—	Risk of Overfitting: Large parameter count increases overfitting risk with small datasets.

Example: Application of BiLSTM to Arabic NER

The attached image illustrates the architecture of a Bidirectional Long Short-Term Memory (BiLSTM) network used in the Named Entity Recognition (NER) task, applied to the Arabic sentence: "زار خليفة حفتر سرت" which means in English (Khalifa Haftar visited Sirte).

1. Input Representation

Before being processed by the LSTM network, each word in the sentence is converted into a rich vector representation. The inputs at each time step (t) consist of:

- **Word Embedding:** A vector representing the semantic and contextual meaning of the word (e.g., using Word2Vec or FastText).
- **Character Representation:** A vector typically generated by a small Convolutional Neural Network (CNN) or LSTM that processes the constituent characters of the word. This helps the model understand the morphological structure of the word and handle **Out-of-Vocabulary (OOV)** words.

These two vectors are concatenated to form the final input vector fed into the BiLSTM layer.

2. The Core Algorithm: BiLSTM Network

The network consists of two parallel layers that capture context from both directions of the sentence.

Table 4.2: Contextual Functionality of Forward and Backward LSTM in BiLSTM for Arabic Named Entity Recognition

Component	Direction	Function	Contextual Significance
Forward LSTM	Right-to-Left (in Arabic script)	Processes the forward sequence (t_{i-1},t_{i-2}), retaining **preceding context**.	Helps determine that "Haftar" follows "Khalifa," the start of a compound name.
Backward LSTM	Left-to-Right	Processes the backward sequence (t_{i+1},t_{i+2}) retaining **following context**.	Helps determine that "Khalifa" precedes "Haftar," the complementing part of the name.

Concatenated Output Vector:

At each time step, the outputs of both LSTM layers, hForward and hBackward, are concatenated to form the final output vector yt. This vector yt carries information about the word **enhanced by the complete context** (preceding and following), which improves classification accuracy.

3. Output and Classification Layer (Output Tagging)

The concatenated vector yt for each word is passed through a **Dense Layer** and then a **Softmax** layer (or **CRF**) to classify it using the **BIO-Tagging** scheme [29].

Table 4.3 Example of BIO Tagging for Arabic Named Entity Recognition

Word	Final Output Tag	Tag Interpretation	Explanation
زار (Visited)	O	**O (Outside)**	The word is a verb and does not belong to any Named Entity (Person, Location, etc.).
خليفة (Khalifa)	B-PER	**B-PER (Begin-Person)**	The word is the *beginning* of an entity representing a **person**.
حفتر (Haftar)	I-PER	**I-PER (Inside-Person)**	The word is *inside* (part of) the same **person** entity that started with "Khalifa."
سرت (Sirte)	B-LOC	**B-LOC (Begin-Location)**	The word is the *beginning* of an entity representing a geographical **location**.

Figure 4.3 provides a detailed illustration of a bidirectional LSTM (BiLSTM) architecture applied to a named entity recognition (NER) task. As shown in the figure, each input word is represented using both word embeddings and character-level representations, which are fed into the BiLSTM network. The forward LSTM processes the sequence from left to right, while the backward LSTM processes it from right to left, enabling the model to capture contextual information from both preceding and succeeding words. The combined hidden states are then used to predict the corresponding entity labels following the BIO tagging scheme (e.g., B-PER, I-PER, B-LOC, O), allowing accurate identification of entity boundaries within the input sequence

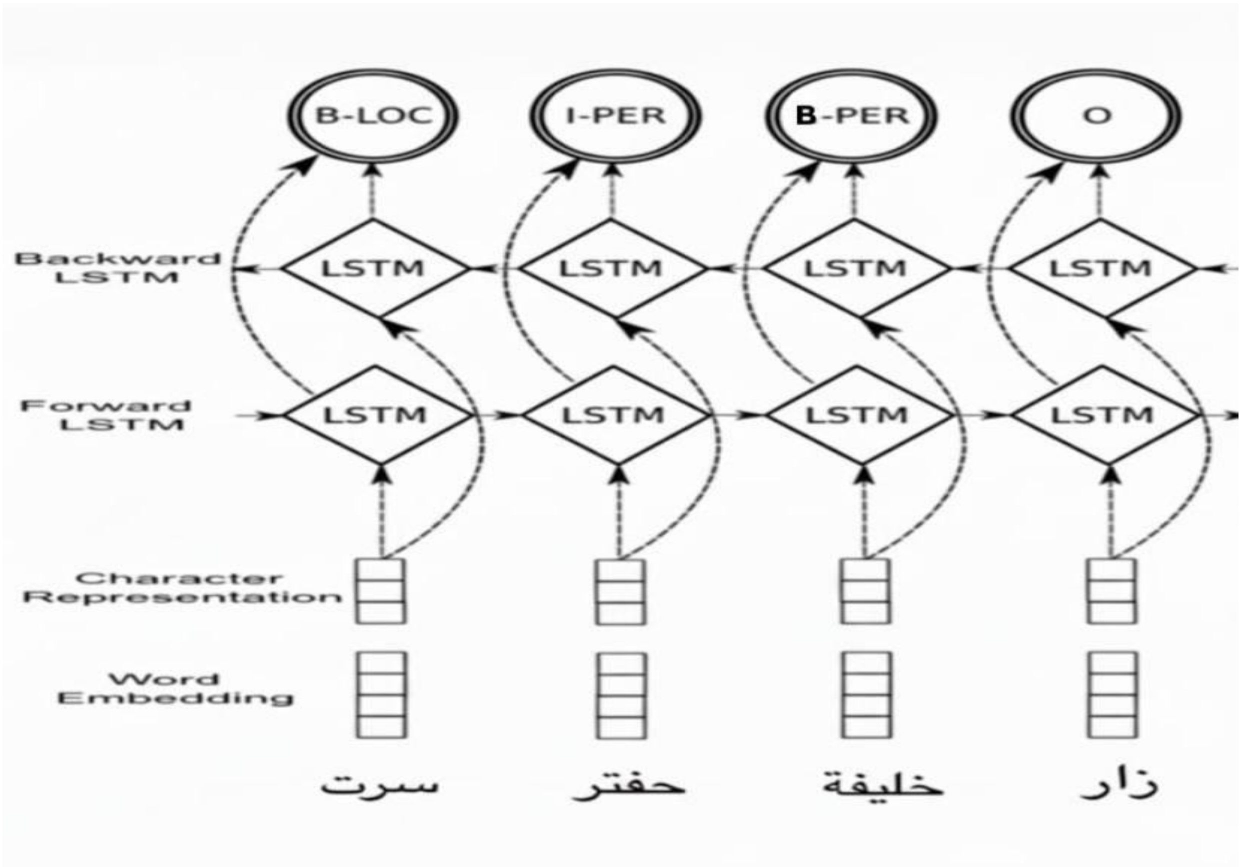

Figure 4.3 Bi-LSTM

4.4.2 Convolutional Neural Networks (CNNs) for Character-Level Features

CNNs are frequently used in NER systems to extract sub word and character-level features, which is especially beneficial for morphologically rich languages. CNNs can capture:

- prefixes, suffixes, and morphological patterns
- root variations
- spelling variations
- out-of-vocabulary (OOV) words

When combined with BiLSTM, CNN-based character embedding significantly boosts NER accuracy.

4.4.3 BiLSTM-CRF Model

The BiLSTM-CRF architecture is one of the most successful models for NER. It consists of:

1. **BiLSTM layers** for contextual encoding
2. **CRF (Conditional Random Field)** as the output layer

The CRF ensures valid label sequences (BIO scheme) and eliminates inconsistencies such as predicting **I-PER** without a preceding **B-PER**.

This architecture has been widely adopted in Arabic NER research due to its robustness and high accuracy.

4.4.4 Transformer-Based Models

Transformers represent the latest breakthrough in NLP through their **Self-Attention** mechanism, which captures long-range dependencies more effectively than recurrent models.

Common Transformer-based models include:

- BERT
- RoBERTa
- DistilBERT
- ELECTRA

Transformers outperform BiLSTM-based systems in most NER benchmarks

4.4.5 The AraBERT Model

Arabic AraBERT is a Transformer-based language model that is specially adapted to Arabic Natural Language Processing (NLP). It expands the original BERT structure with the addition of linguistic and morphological attributes peculiar to the Arabic language. With its large corpus of Arabic language training and its tokenisation approach optimised to do so, AraBERT achieves state-of-art performance in a range of Arabic NLP tasks, such as Notably, Named Entity Recognition.

4.4.5.1 Technical Mechanism

1. **Pre-training:**

- Masked Language Modeling (MLM): Predict masked tokens using bidirectional context.
- Next Sentence Prediction (NSP): Determine logical sentence ordering.

2. **Fine-tuning:**

- Task-specific classifiers (e.g., for NER) are added on top of the base model.
- Base weights are slightly adjusted to improve task performance with limited labeled data.

4.4.5.2 NER with AraBERT

1. Input: Tokenized Arabic text.
2. Contextual Embeddings: Each token mapped to a contextual vector.
3. Classification: Linear layer predicts entity categories (PERSON, LOC, ORG).
4. Advantage: Resolves ambiguities using bidirectional context (e.g., distinguishing "القاهرة" as a city vs. adjective).

4.4.5.3 Mathematical Foundations of AraBERT

The transformer model (The Modern Architecture)

This image represents the architectural structure of the Transformer model, the foundation for modern models like BERT and GPT. This model has revolutionized the field of Natural Language Processing (NLP).

Key Components:

- **The Encoder (Left Side):**
  - It receives the Inputs and the Input Embedding's.
  - Positional Encoding: This component is added to enable the model to understand the order of words in the sentence, a function that RNN/LSTM networks handled automatically due to their sequential nature.
  - Multi-Head Attention Mechanism: This is the core of the Transformer. It allows the model to "weigh" the importance of all other words in the sentence when processing

a specific word. Unlike LSTM, which relies on local memory, the Transformer can capture long-range dependencies instantaneously.

- Add & Norm (Addition and Normalization) and Feed Forward Network: These layers process the outputs and enhance training stability.

- **The Decoder (Right Side):**
  - It receives previous Outputs (during training, the sequence is shifted to the right).
  - Masked Multi-Head Attention: This allows the Decoder to only attend to the words that have been generated before it in the sequence, ensuring that generation occurs in a sequential order (as in sentence generation).
  - Encoder-Decoder Attention Layer: This allows the Decoder to "look" at the Encoder's outputs and extract the relevant information for the word currently being generated.
- **Output Layers:** A **Linear** layer is followed by a **Softmax** layer to convert the vectors into **Output Probabilities** for the next word in the vocabulary.

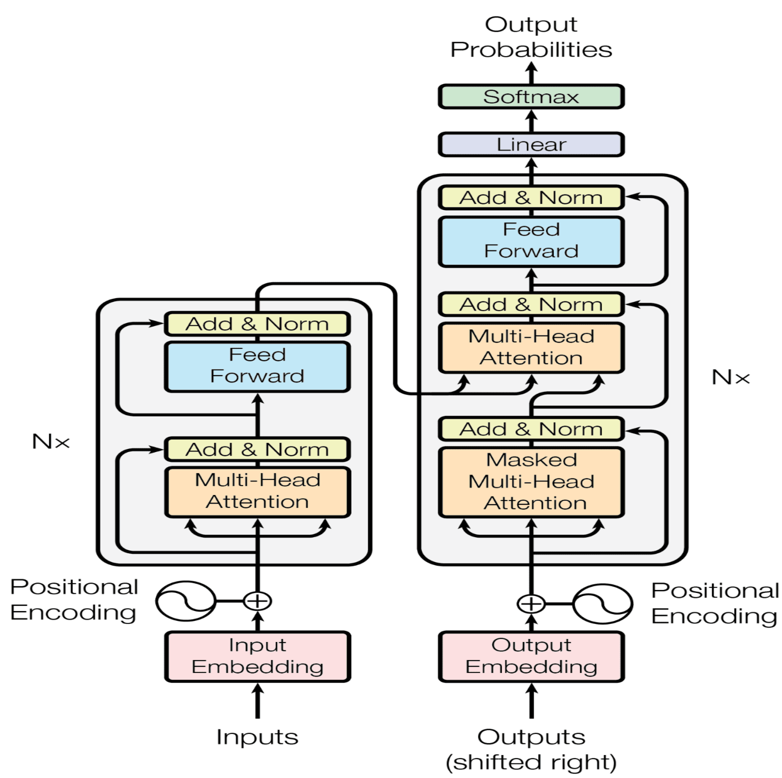

Figure 4.4 BERT model[56]

Mathematical Foundations of AraBERT

The AraBERT model is built based on Transformer architecture, first released by Google in the year 2017 and more specifically on the Bidirectional Encoder Representations from Transformers (BERT) platform. The operations of these models are based on the following basic mathematical formulae:

. **Multi-Head Attention**

This is the core mechanism in Transformer that allows the model to dynamically weigh the importance of different words in a sentence.

Scaled Dot-Product Attention:

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

where:

- Q (Queries), *K* (Keys), and V (Values) are matrices derived from the input embedding.
- *dk* is the dimension of the key vectors.

Multi-Head Attention:

$$\text{head}_i = \text{Attention}\left(QW_i^Q,\, KW_i^K,\, VW_i^V\right), \qquad \text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)\, W^O$$

where:

- w_i^Q, w_i^k, w_i^v are trainable weight matrices for each head.
- *WO* is the output weight matrix.

2. **Layer Normalization**

Used to stabilize the training process:

$$\text{LayerNorm}(x) = \gamma \odot \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta$$

where:

- μ is the mean, σ^2 is the variance.
- γ, β are trainable parameters.
- ϵ is a small constant for numerical stability.

3. **Feedforward Neural Network**

Each layer in BERT contains this network:

$$\mathrm{FFN}(x) = \max(0,\, xW_1 + b_1)\, W_2 + b_2$$

This is a **ReLU** activation function between two linear layers.

4. **Contextual Word Representations**

The final representation for each word is computed by combining the multi-head attention output with the original input (Residual Connection) followed by normalization:

$$Z \; = \; \mathrm{LayerNorm}(X + \mathrm{MultiHead}(Q, K, V))$$

5. **Training Objectives for BERT/AraBERT**

a. **Masked Language Modeling (MLM)**:

A percentage of words (e.g., 15%) are masked and replaced with [MASK], and the model learns to predict the original word:

$$P(w_{\text{masked}} \mid \text{context}) = \mathrm{softmax}(W\, h_{\text{masked}} + b)$$

b. **Next Sentence Prediction (NSP)**:

The model learns to determine whether sentence BB logically follows sentence AA:

\$ $P(\text{IsNext} \mid A, B) = \sigma\big(W \cdot h_{[\text{CLS}]} + b\big)$ \$

where:

- $h_{[CLS]}$ is the representation of the special [CLS] token.
- σ is the Sigmoid function.

6. **Total Loss**:

$$\mathcal{L} = \mathcal{L}_{\mathrm{MLM}} + \mathcal{L}_{\mathrm{NSP}}$$

Key Notes:

- $\mathcal{L}_{\mathrm{MLM}}$

The loss for the Masked Language Modeling (MLM) task, typically computed using Cross-Entropy over the masked tokens only.

- $\mathcal{L}_{\mathrm{NSP}}$

The loss for the Next Sentence Prediction (NSP) task, computed using Binary Cross-Entropy based on the probability that the second sentence follows the first.

- Joint Training

The model learns both tasks simultaneously, improving contextual understanding at both the token and sentence levels.

- Default Weighting

In the original BERT setup, no additional weighting is applied both losses are simply summed.

- However, custom setups may assign different weights to each component if needed
- AraBERT reuses the same mathematical foundations as the original BERT model however, it is specifically trained on Arabic texts (including dialects).
- The model supports **bidirectionality** due to the Masked Language Modeling mechanism.
- These equations are applied to each token in the sentence in parallel (thanks to the parallelism in Transformer).

Table 4.4: AraBERT Algorithm: Advantages and Disadvantages Sumary

Advantages	**Disadvantages**
Deep, Bidirectional Context Understanding: Processes inputs using the Transformer architecture for a comprehensive grasp of context (past and future words).	High Computational Complexity: Pre-training and inference demand vast computational resources (powerful GPUs/TPUs and large memory).
Effective Arabic Adaptation: Pre-trained on a massive (77 GB) Arabic corpus, ensuring a strong grasp of linguistic and morphological features.	Maximum Sequence Length Limitation: Limited to a maximum input length (typically 512 tokens), restricting its ability to process very long documents directly.
Optimized Subword Tokenization: Uses an Arabic-specific Subword Tokenization scheme to efficiently handle the rich derivation and morphology of the language.	Potential Data Bias: The quality and sources of the training data may introduce unwanted biases (political, social, or dialectal) into the model.
State-of-the-Art Performance: Significantly outperforms global, non-Arabic-specific models on Arabic NLU tasks (NER, Classification, QA).	Suboptimal for Some Dialects: Primarily trained on Modern Standard Arabic (MSA), leading to less efficient performance on some Arabic colloquial dialects.
Flexibility (Transfer Learning): Can be easily fine-tuned on smaller, task-specific Arabic datasets, accelerating specialized model development.	Resource-Intensive Fine-Tuning: Even the fine-tuning process can require significant resources compared to classical models.

4.6 Summary

In this chapter, the natural language processing (NLP) and the significance of it in the process of allowing the machines to understand and produce human language has been thoroughly examined. It has discussed the key elements of NLP and highlighted necessities of tokenization, morphological analysis, syntactic parsing, semantic representation, and sequence labeling. Special emphasis was put on the issues that relate to the Arabic processing, namely, the rich morphology, the diversity of its orthography and the variety of dialects.

The chapter is devoted to the LNPS task of Named Entity Recognition (NER), which is a fundamental sequence-labeling problem. It outlines the theoretical foundations of NER, demonstrates how it has been applied in areas such as information retrieval, sentiment analysis and information mining, and the challenges encountered during executions of the process especially in dialectic as well as noisy social media.

The different deep-learning structures typically used to learn NER are discussed (including Recurrent Neural Networks (RNNs), Long Short-term Memory (LSTM) networks, and finally the detailed description of the Bidirectional LSTM (BiLSTM) structure. The mathematical modeling, structural implementation, and benefits and weakness of the BiLSTM are described, thus illustrating its capability of acuity in long-range association and bidirectionality in context that are invaluable in entity classification with high-level Arabic text.

A comprehensive analysis of the AraBERT model, a Transformer-based model, which has already been pre-trained specifically on Arabic, is finally used. The technical processes, which are: Masked Language Modeling (MLM), Next Sentence Prediction (NSP), multi -head attention and feed -forward layers are explained along with the mathematical equation involved. The strong points of the model (including the rich bi-directional contextual representation and the adaptation to the Arabic morphemes) are counterbalanced with the computational complexity and the difficulty of using some of the under-resourced dialects.

The combined chapter provides the vigorous conceptual and technical groundwork of the methodological and experimental elements that could be used later in other chapters. The lessons learned in the paper highlight the importance of combining the Transformer-based contextual embedding with sequential models to thoroughly face the linguistic properties of the modern standard Arabic and the regional dialects of Arabic, and thus, provide the means of creating strong NLP applications that will foster the understanding of the Arabic language and broaden the understanding of the linguistic process in many different directions.

Chapter Five: Methodology

5.1 Introduction

In a sense, this chapter outlines a very strict framework of the design of a complex Arabic Named Entity Recognition (NER) system, which is specifically designed to suit the Libyan dialect. The key issue that will be addressed in this thesis is how to ingest and process copious amounts of informal textual information derived through social media, especially Twitter, and derive topics of substantive interest to the Libyan sociolinguistic situation. Due to the complex nature and unformatted quality of such data, the strategy predicts emerging and developed deep-learning and transfer-learning methods to become essential elements in the attainment of strong performance in the models.

The harvesting of approximately four thousand tweets was done directly through the Twitter API. After a tedious preprocessing pipeline consisting of careful cleaning, tokenization, and proper feature engineering the final dataset was partitioned into an 80 percent training set and 20 percent test set. This division is an equilibrium point of having enough data to learn the model effectively and maintaining a dependable evaluation of the generalization ability.

5.2 Data Collection and Libyan Corpus Development

Despite the effectiveness of deep-learning models, the most important factor is the integrity and comprehensiveness of the corpus, especially in cases where the model is expected to learn a low-resource dialect, such as the Libyan variety. To address this issue, we employed a custom-built data-collection and annotation workflow, defined by the following steps:

5.2.1 Data Collection Strategy

The textual data were mined off the Twitter (X) platform due to its massive use of the Libyan dialect in informal, every day and conversational settings. To achieve a wide geographical and demographical coverage, we selected the list of local news versions and online community stories that include a wide range of media outlets, including, and not confined to the following:

- **General News Sources:** Such as (@Ain_Libya), (@Huna_Alasima), and (@Alunwan_Newspaper).

- **Regional Pages:** Covering different areas (East, West, and South) to capture the diversity in vocabulary and emerging entities.

The data collection process spanned a period from the beginning of 2024 to mid-2025.

5.2.2 Preliminary Corpus Statistics

The total volume of collected data, prior to annotation, was as follows:

Table 5.1 Preliminary Corpus Statistics

Metric	Value
Total Tweets Collected	Approximately 4,000 tweets
Total Number of Tokens	38,747 tokens
Collection Timeframe	January 2024 – June 2025
Platform Used	Twitter (X)

5.3 Data Preprocessing and Sanitization

Prior to annotation, the collected data underwent several essential cleaning steps to prepare it for the machine learning process:

1. **Removal of Symbols and Hashtags:** All emoji's, URLs, hashtags, and user mentions (@Mentions) were removed.
2. **Whitespace Standardization:** Excessive whitespaces and delimiters were standardized to ensure consistency in the input data.
3. **Storage:** The cleaned corpus was stored in an Excel and CSV file format to streamline the subsequent manual annotation process.

5.4 Manual Annotation Methodology

The manual annotation process is the critical step for transforming unstructured text data into a labeled training corpus for the NER model.

1. **Annotation Approach:** The data was labeled using an entity-level tagging approach, where each word or word sequence representing a named entity was identified and categorized under the appropriate class.
2. Entity Categories: The named entities in the corpus were divided into specific categories for classification purposes, including: Persons (Person), Locations (Location), and Organizations (Organization). The annotation process was carried out at the token level using the BIO (Begin–Inside–Outside) or BILOU (Begin–Inside–Last–Outside–Unit) tagging scheme. This approach enables precise identification of entity boundaries and clearly distinguishes tokens that do not belong to any entity
3. Annotation Statistics

Table 5.2 Annotation Statistics

Classification	**Count**
B-ORG	973
I-ORG	1468
B-LOC	1411
I-LOC	868
B-PERS	969
I-PERS	770
O	32,267
Total	38,726

4. **Aggregated by entity:**

Table 5.3 Entity Distribution in the Annotated Corpus

Entity	**Count**
ORG	2,441
LOC	2,279
PERS	1,739
O	32,267

5.5 Steps of Machine Learning Algorithms for NER

The system development involves three fundamental stages:

1. **Initialization Step:** Dataset preparation, preprocessing, and environment setup
2. **Learning Step:** Model training on annotated data to learn contextual representations
3. **Evaluation Step:** Assessing model performance using Precision, Recall, F1-Score.

Figure 5.1 illustrates the full pipeline from raw data to final model evaluation

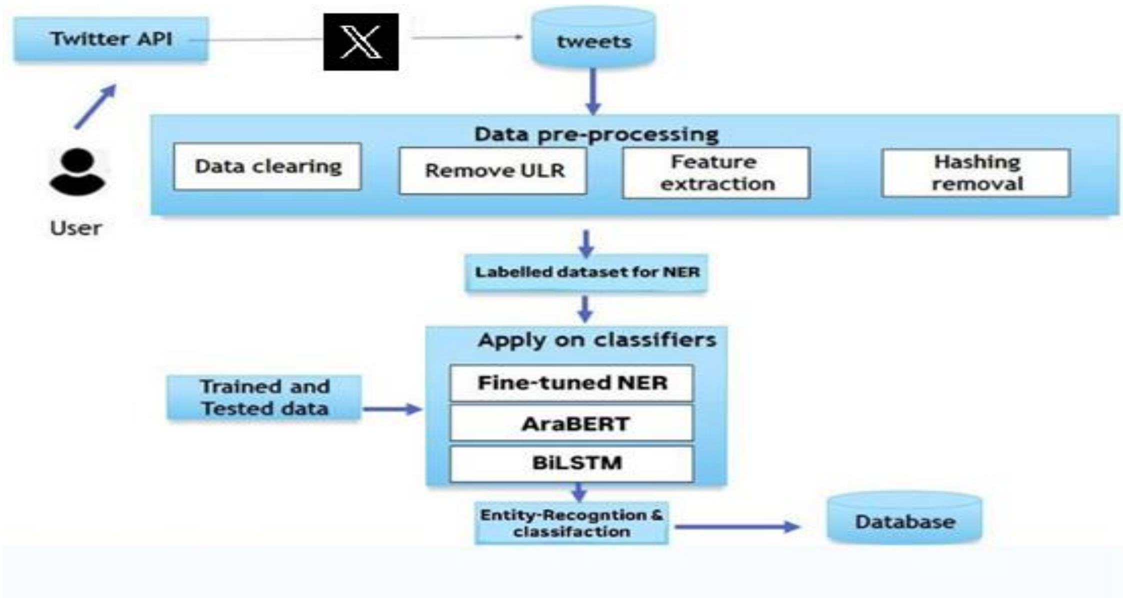

Figure 5.1 steps of Machine Learning Algorithms for NER

5.6 Model Architecture

The architecture of the Named Entity Recognition (NER) system combines sequence modeling and contextual embedding to effectively capture the nuances of the Libyan dialect in Arabic. Two main model types were adopted:

5.6.1 BiLSTM (Bidirectional Long Short-Term Memory)

- **Purpose:** To capture the bidirectional context of words in a sequence, meaning the model considers both previous and subsequent words when classifying each token.
- **Mechanism:**
  - Processes the input sequence in two directions: forward (left-to-right) and backward (right-to-left).
  - Memory gates in LSTM cells allow the model to store long-term dependencies and avoid vanishing gradient issues.
- **Output Layer:**
  - A Dense Layer followed by a Softmax Layer predicts the probability distribution for each token over the NER categories (B-PER, I-LOC, O, etc.).

Advantages:

- Efficient for smaller datasets.
- Captures sequential dependencies in text.

5.6.2 AraBERT / Transformer-based Models

- **Purpose:** To leverage pre-trained contextual embedding for Arabic text, capturing rich semantic and syntactic information from large corpora.
- **Mechanism:**
  - Each token is transformed into a contextualized vector representation, reflecting its meaning based on surrounding words.
  - Multi-Head Attention is applied, allowing the model to weigh the importance of all other tokens in the sentence.
- **Fine-Tuning:**
  - The pre-trained AraBERT model is fine-tuned on the manually annotated Libyan corpus for the NER task.
- **Output Layer:**
  - Similar to BiLSTM, uses Dense + Softmax for token-level classification.

Advantages:

- Captures complex language patterns and context.
- Handles ambiguity in multi-word named entities.
- Particularly effective in low-resource dialects due to transfer learning.

5.6.3 Comparative Notes

Table 5.4 Comparison between BiLSTM and AraBERT Models

Feature	**BiLSTM**	**AraBERT**
Context	Sequential, bidirectional	Deep, multi-head attention
Pre-training	Trained from scratch	Pre-trained on large Arabic corpora
Performance on NER	Moderate	High, especially on complex, low-resource dialects
Dataset Size	Works on smaller datasets	Benefits from fine-tuning, even with moderate data

5.7 Annotation and Training

The manual annotation of the collected data is a critical step in creating the "Gold-Standard" necessary for supervised machine learning.

5.7.1 Entity Classes and Tagging Scheme Definition

The Named Entity categories were defined based on their significance within the Libyan media and social context. Three main classes were adopted:

1. **Persons (PER):** Includes names of individuals and public figures.
2. **Organizations (ORG):** Includes names of companies, governmental institutions, and non-profit organizations.
3. **Geographical Locations (LOC):** Includes names of cities, regions, and various geographical sites, with particular attention paid to representing the diverse Libyan regions.

To facilitate the identification of multi-word entity boundaries, the BIO (Beginning, Inside, Outside) tagging scheme was applied:

- **B-tag:** Indicates the beginning of a named entity.
- **I-tag:** Indicates subsequent words falling within the scope of the same named entity.
- **O-tag:** Indicates words that do not belong to any named entity.

term	classification
الزهرة	O
من	O
ضمن	O
محطات	O
اللي	O
شركة	B-ORG
البريقة	I-ORG
منزلينه	O
يفتح	O
24	O
ساعه	O
ومن	O
امس	O
موقف	O
ليبيا	B-LOC
في	O
النظام	B-ORG
المالي	I-ORG
الدولي	I-ORG
الصديق	B-PERS
الكبير	I-PERS
ينصب	O
خياما	O
لاستقبال	O

Figure 5.2 Example of Applying the BIO Tagging Scheme for Identifying Multi-Word Named Entity Boundaries

Table 5.5 Distribution of Named Entities by Classification

classification	Count
ORG	2441
LOC	2279
PERS	1739
O	32267

Table 5.6 Distribution of BIO Tags Across Named Entity Types

Classification	Count
B-ORG	973
I-ORG	1468
B-LOC	1411
I-LOC	868
B-PERS	969
I-PERS	770
O	32267
Total	**38726**

5.7.2 Model Training and Deep Learning Adoption

At the end of the process of the manual annotation take place, the completely labelled corpus has been utilized as the major training data. The scholarship uses modern deep-learn algorithms, and it encompasses the state of the art when it comes to handling the complex language forms characteristic of the Arabic and, more to the point, the Libyan dialect.

The models used to perform the Named Entity Recognition (NER) task were carefully constructed to perform an extraction of linguistic and contextual features in a rather efficient manner, thus being able to guarantee proper classification of the named entities on the basis of the annotated input.

Table 5.7 Example of Annotated Input Using the BIO Tagging Scheme for NER

Column1	Column 2
الأهلي	B-ORG
بنغازي	I-ORG
يعبر	O
الأندلس	B-LOC
بثلاثية	O
ويتأهل	O
إلى	O
دور	O
الـ16	O
من	O
كأس	B-ORG
ليبيا	I-ORG
معبر	B-LOC
راس	I-LOC
اجدير	I-LOC
الحدودي	I-LOC

5.7.3 Procedural Steps of the Named Entity Recognition (NER) Algorithm

1. Phase 1: Input and Pre-processing

The algorithm begins by transforming the raw text into a format process able by the model:

- **Tokenization:** The input sentence or text is segmented into small linguistic units (**Tokens**), which may be words or sub-words. This is the initial step for feeding data into the model.
- **Word Embedding:** Each token is converted into a dense numerical vector (Vector Representation). This vector represents the word's meaning and context within the language. In models like AraBERT, these are Contextualized Embedding, meaning the word's representation changes based on the sentence it appears in.

2. Phase 2: Feature Extraction and Context Modeling

In this phase, the Model works to understand the bidirectional context of the word within the sentence:

- **Context Modeling:**
  - **In BiLSTM Models:** The cell receives the sequence in two directions (forward and backward), utilizing its mathematical gates to control information flow and memory storage. This produces a rich representation combining the past and future context for each word.
  - **In AraBERT/Transformer Models:** The Multi-Head Attention mechanism is applied in parallel. This mechanism allows each word to identify and weigh the importance of all other words in the sentence, generating a deep contextual representation for the word.
- **Final Representation Generation:** At the end of the BiLSTM or Transformer layers, an enhanced contextual vector is obtained for each word. This vector now encapsulates all necessary information about the word, including its context and its potential role as a named entity.

3. Phase 3: Classification and Tagging

This is the step where the algorithm makes its final decision about the identity of each word:

- **Classification Layer:** The final contextual vector for each word (from Phase 2) is passed through a Dense Layer or a Softmax Layer.
- **Tag Prediction:** The classification layer outputs a probability distribution over all defined Named Entity categories (e.g., B-PER, I-LOC, O).

  - The tag with the highest probability is selected for each word.
  - A tagging scheme like BIO (Beginning, Inside, Outside) is used to determine the entity category (PER, LOC, ORG) and define its boundaries (B for start, I for inside, O for outside).

$$\text{Predicted Tag} = \arg Max_k \ \text{P} \ (Tag_k \mid \text{Contextual Vector})$$

4.Phase 4: Output Generation

- **Entity Aggregation:** Words carrying a B-tag followed by I-tags (e.g., B-ORG followed by I-ORG) are grouped together to form the complete Named Entity.
- **Final Output:** The algorithm presents the text along with the extracted and classified Named Entities (e.g., "المصرف [ORG] الأهلي [ORG]" / "Al-Ahli [ORG] Bank [ORG]", "مدينة [LOC] طرابلس [LOC]" / "Tripoli [LOC] City [LOC]").

5.7.4 Summary of basic model building steps for Named Entity Recognition (NER):

Step 1: Collect dataset - by extracting it from Twitter data store. Organize data including deleting and removing unnecessary data.

Step 2: Create our model in Python.

Step 3: Deep analysis and training of multi-label model data. Predict using test dataset and evaluate accuracy.

Step 4: Predict on new dataset.

Let S be **the Named Entity Recognition and Classification System**
$S = \{ Tw, Pt, Er\}$ = Named Entity Recognition System.
Tw = Tweets extracted from Twitter.
Pt = Tweet preprocessing (removal of duplicates, removal of English words, URL, removal of stop words).
$S1$ = Named Entity Recognition using tools.
$Er = \{ Pr, Lc, Org, O\}$

- Pr : **Persons Class**
- Lc: **Locations Class**
- Org: **Organizations Class**
- O: **Other/Outside Class**

Where:

- $Pr = \{ P_1, P_2, \dots\dots, P_n\}$ = **Persons Class**
  - $P_1, P_2, \dots\dots, P_n$ is the class of the set of recognized Person entities (e.g., names of individuals).
- $Lc = \{ L_1, L_2, \dots\dots, L_n\}$ = **Locations Class**
  - $L_1, L_2, \dots\dots, L_n$ is the class of the set of recognized Location entities (e.g., names of cities or countries).
- $Org = \{ O_1, O_2, \dots\dots, O_n\}$= **Organizations Class**
  - $O_1, O_2, \dots\dots, O_n$ is the class of the set of recognized Organization entities (e.g., names of companies or institutions).
- $O = \{ O'_1, O'_2, \dots\dots, O'_n\}$= **Other/Outside Class**
  - $O'_1, O'_2, \dots\dots, O'_n$is the class of the set of recognized Other entities (e.g., dates, products, monetary values, or non-entity words).

5.8 Implementation Environment and Training Configuration

To operationalize the NER models, as well as ensure reproducible experimental results, a set of widely-known deep learning and Natural Language Processing (NLP) libraries were used to configure the training environment. This section outlines the computational paradigm, data preparation protocol and the training conditions that will be used in the investigation.

5.8.1 Working Environment and Libraries Used

This project was implemented in a Python environment, a leading programming language in the fields of machine learning and Natural Language Processing (NLP), thanks to its flexibility and the availability of numerous specialized libraries.

Working Environment and Key Libraries Used

The project was implemented using Python, which serves as the core programming environment due to its flexibility and strong support for Machine Learning and NLP.

Table 5.8 Working Environment and Key Libraries Used

(Category)	(Library)	(Primary Function in the Project)
Deep Learning Framework	**TensorFlow & Keras**	The main framework for building, training, and running complex Deep Learning models (Neural Network Architectures).
NLP Modeling	**Hugging Face Transformers**	Essential for utilizing and fine-tuning state-of-the-art Transformer Models (e.g., BERT) for Named Entity Recognition (NER) in the Libyan dialect.
Data Handling & Processing	**Pandas & NumPy**	Used for efficient data manipulation, reading input files, organizing data into tables/arrays, and high-performance numerical operations.
Utility & Metrics	**Scikit-learn (sklearn)**	Used for critical preprocessing steps like Label Encoding, dividing data into Training/Testing sets (Train-Test Split), and generating performance evaluation metrics (Classification Report).

5.8.2 Model Architecture and Fine-Tuning

The main model of deep learning was *aubmindlab/bert-base-arabertv2* that was implemented using TFAutoModelForTokenClassification. Being a pre-trained Arabic contextual representation, AraBERT can provide rich semantic representations Opinions with which the Libyan dialect was fine-tuned under the supervision on the corpus annotated manually.

The probability distribution obtained with this model is a probability distribution on all classes of BIO entities as represented by a particular token and predictions are made as indicated:

$$\text{Predicted Tag} = \arg Max_k \ \text{P} \ (Tag_k \mid \text{Contextual Vector})$$

This formulation ensures that each contextualized token embedding is mapped to its most probable named entity category.

5.8.3 Data Preparation and Token-Level Alignment

The dataset was loaded from the annotated Excel file (LastData.xlsx) and underwent several preparation stages:

- **Normalization:** Removal of diacritics to eliminate orthographic variation.

- **BIO Tag Encoding:** All entity tags (e.g., B-ORG, I-LOC) were encoded using Label Encoder.
- **Dataset Split:** The fully annotated corpus was divided into 80% for model training and 20% for final evaluation, consistent with the methodology outlined in Section 5.7.
- **Tokenization:** Utilizing AraBERT's tokenizer with a maximum sequence length of **64 tokens**.
- **Label Alignment:** A specialized alignment function ensured that:
  - Only the first subword (the head token) of each word retained its actual BIO label.
  - All other subword segments and special tokens were assigned –100, ensuring they are excluded from gradient updates.

This alignment step is critical in transformer-based NER systems to avoid artificially inflating accuracy or misrepresenting entity boundaries.

5.8.4 Training Configuration and Optimization Strategy

The training process was governed by the following hyper parameters and optimization settings:

Table 5.9 Hyperparameter and Optimization Settings for Model Training

Setting	**Value / Description**
Epochs	6
Batch Size	16
Learning Rate	**2×10^{-5}** (AdamW optimizer)
Scheduler	Learning Rate Warmup (10% of total steps)
Loss Function	Masked Sparse Categorical Crossentropy (ignores –100 labels)
Metric	Masked Accuracy (evaluates only valid token labels)
Callbacks	ModelCheckpoint, EarlyStopping (patience = 3)

The loss and accuracy functions were adapted to operate only on genuine label-bearing tokens, ensuring a reliable assessment of model learning.

5.8.5 Evaluation Procedure

A 20 per cent held-out test set was used to evaluate the model performance through the same tokenization and alignment procedures used during the training. The labels that received the value of -100 were masked in case they contaminated the results in the evaluation.

A comprehensive token-level performance report was generated using:

- **Precision**
- **Recall**
- **F1-Score**

computed via sklearn.metrics.classification_report, offering granular insights into both individual BIO tags and aggregated entity types (PER, ORG, LOC)

5.9 Summary

In Named Entity Recognition (NER), the 'O' (Non-Entity) class is overwhelmingly dominant. Therefore:

- **Accuracy** is typically disregarded as it is artificially high due to the huge number of True Negatives (TN).
- **Precision** and **Recall** are used individually to understand the model's strengths and weaknesses.
- **F1-Score** is adopted as the **primary and most reliable metric** for assessing
- the overall performance of an NER model.

Detailed Translation of Key Training Settings

The main training settings are defined in the "Settings and Paths" section of the code and are used later when compiling the model:

Table 5.10 Detailed Description of Key Training Hyperparameters and Settings

Parameter (Arabic)	Code Variable	Specified Value	Explanation (English)
(Learning Rate - LR)	LEARNING_RATE	$\text{Learning Rate (LR)} = 2 \times 10^{-5} (0.00002)$	A small value (typically between (1×10^{-5} and 5×10^{-5}) essential for fine-tuning pre-trained Transformer models to avoid drastic weight changes.
(Batch Size)	BATCH_SIZE	$\text{BATCH\_SIZE} = 16$	The number of samples (words/sentences) processed in one training step before the model weights are updated.
(Epochs)	EPOCHS	$\text{EPOCHS} = 6$	The number of times the entire training dataset passes through the model.
(Loss Function)	masked_loss (based on SparseCategorical Crossentropy)	$L = -\sum_i Mask_i . \log(P\ (y_i\ \lvert x_{i)})$	Designed to ignore masked tokens where ($Mask_i = 0$) for ignored tokens.
(Optimizer)	optimizer (created by create_optimizer)	AdamW (with Learning Rate Scheduler)	An Adam variant that includes **Weight Decay**. It is created using transformers.create_optimizer, which implements a **Learning Rate Scheduler** with a **Warmup** period for smooth training start.

Additional Explanation of the Optimizer

Lastly, the architecture makes use of the create optimize function found in transformers library which generates an optimizer known as AdamW (Adam with weight decay). This optimizer is still the default when performing fine-tuning of Transformer models including BERT and AraBERT:

- **Adam**: Efficiently adapts the learning rate for each parameter.
- **Weight Decay (Regularization)**: Acts as a Regularization mechanism to prevent Overfitting by minimizing large weights, which is crucial when using pre-trained models.

Table 5.11 Transformer Model Training Configuration and Key Numerical Values

Setting	**Calculated Value**
Total Training Steps (num_train_steps)	Total Training Steps =num_batches_per_epoch×EPOCHS
Warmup Steps (num_warmup_steps)	Warmup Steps =num_train_steps//10(10% of total steps)

This setup ensures that the learning rate starts at zero, gradually increases to the maximum value (2×10^{-5}) during the Warmup steps, and then gradually decreases until the end of training.

Interpretation of Key Numerical Values

Table 5.12 Key Numerical Hyperparameters and Their Purpose

Value	**Context in Code**	**Interpretation / Purpose (English)**
-100	aligned_labels	Masking Value: Used to tell the loss function to ignore padding and subsequent word tokens during training.
2×10^{-5}	LEARNING_RATE	Learning Rate: A small value for fine-tuning the pre-trained AraBERT model.
16	BATCH_SIZE	Batch Size: The number of samples the model is trained on in one step.
6	EPOCHS	Number of Epochs: The number of full passes over the dataset.
0.2	TEST_SPLIT_SIZE	Test Split Ratio: 20% of the data is allocated for testing.
64	MAX_LENGTH	Maximum Sequence Length: The maximum number of tokens permitted per input sequence.

Chapter Six:
Results and Discussion

6. Result and Discussion

A close discussion of the findings of the Arabic Named Entity Recognition (NER) task classifier involving AraBERT model and BiLSTM model are provided in this chapter. The discussion will focus on the performance of the models, entity-related results, difficulties, and main observations based on Chapter 5.

6.1 Overall Performance Analysis

The models were evaluated using standard metrics: Accuracy, Precision, Recall, and F1-Score. Table 6-1 summarizes the aggregate performance of both models.

Table 6.1 Summary of Overall Performance Comparison Between AraBERT and BiLSTM

Metric	AraBERT Result	BiLSTM Result	Key Finding
Accuracy	0.92	0.94	**BiLSTM** is more accurate overall.
Macro Avg Precision	0.80	0.86	**BiLSTM** is more precise across all classes.
Macro Avg Recall	0.79	0.78	**AraBERT** has slightly better recall across all classes.
Macro Avg F1-Score	0.79	0.82	**BiLSTM** achieves better overall balanced performance.
Weighted Avg F1-Score	0.92	0.93	**BiLSTM** performs slightly better when weighting by sample size.
Total Support (Total Samples)	7821	7770	(Total number of tokens evaluated)

Key Findings:

- The BiLSTM model was the most successful in terms of general Accuracy of 0.94 and Macro Avg F1 -Score of 0.82, meaning that it performs best of all entity types without being over-optimistic with the majority class (O).
- The AraBERT performed slightly higher on Macro Avg Recall (0.79), i.e. it was slightly better at recognizing all entities relevant to the entire dataset.

6.2 Detailed Performance Analysis by Entity Category

Table 6-2 details the performance of each model across the four Named Entity categories: **Location (LOC)**, **Person (PERS)**, **Organization (ORG)**, and **Outside Entity (O)**.

Table 6.2 Detailed Comparison of Performance Metrics Results by Entity Category

Category	**Model**	**Precision**	**Recall**	**F1-Score**	**Support**
LOC	AraBERT	0.76	0.77	0.77	483
LOC	BiLSTM	0.85	0.77	0.81	457
ORG	AraBERT	0.60	0.58	0.59	486
ORG	BiLSTM	0.71	0.65	0.68	463
PERS	AraBERT	0.89	0.83	0.86	381
PERS	BiLSTM	0.94	0.74	0.83	322
O	AraBERT	0.95	0.96	0.96	6471
O	BiLSTM	0.96	0.98	0.97	6528

Detailed Observations on Entity Performance

1. **Organization (ORG) Category:** This was the poorest category in the both the models though BiLSTM improved considerably, the F1-Score rose by 0.59 (AraBERT) to 0.68. This implies that BiLSTM is much more efficient in differentiating wide technological borders of organizations.
2. **Location (LOC) Category:** BiLSTM obviously performed better than AraBERT, F1 - Score=0.81 (as compared to AraBERT=0.77), with the help of high Precision (0.85).
3. **Person (PERS) Category:** AarBERT has the maximum balanced performance of F1-Score of 0.86 due to its high Recall (0.83). On the other hand, BiLSTM had very high Precision (0.94) though lower Recall (0.74), meaning that it was much more conservative and had more actual Person entities misdirected.

6.3 Graphical Representation of Performance Results

Figure 6.1 provides a visual comparison of the F1-Scores for both models across the four entity categories, reinforcing the observed trade-offs and strengths.

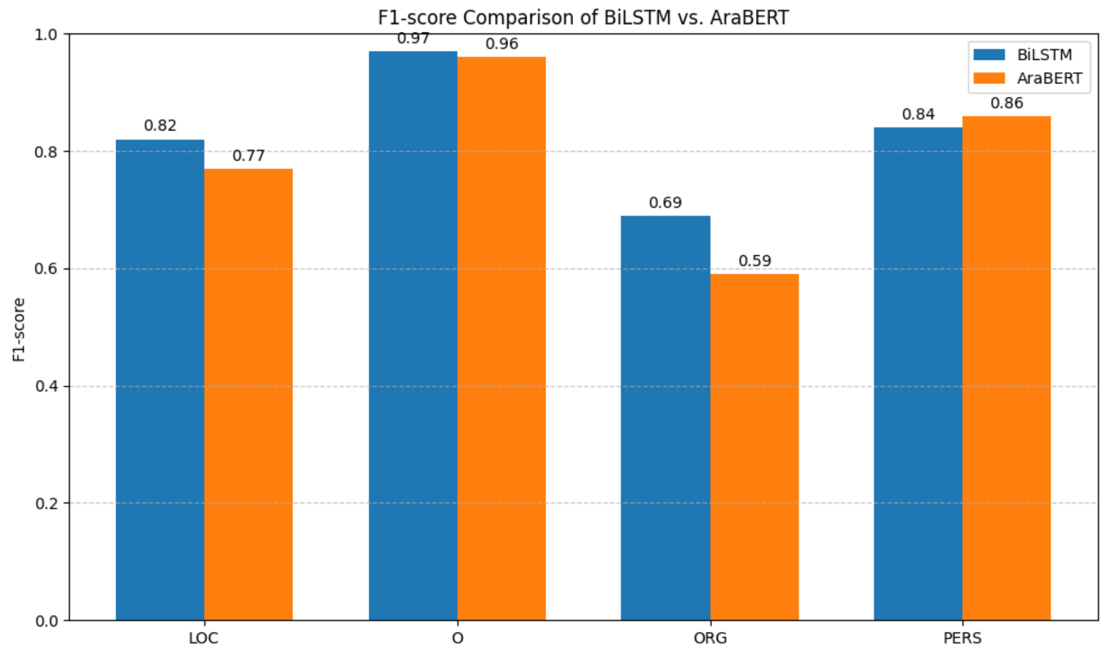

Figure 6.1 F1-Score Comparison of AraBERT and BiLSTM Models by Entity Category

Examples of Named Entity Recognition (NER) Classification Results

Examples of sentences that have been fed through the Named Entity Recognition model are presented in the tables below, and show the token (word), the end entity label, and its confidence in the classification.

Tagging Notes:

- **O:** Outside Entity (Non-entity word).
- **B-LOC:** Beginning of a Location Entity.
- **B-ORG:** Beginning of an Organization Entity.

1. Results of the First Example (Sentence including "Ghat")

Table 6.3: Results of the First Example (Sentence including "Ghat")

Token	**Final Tag**	**Confidence**	**Notes**
ثلاثة (three)	O	0.99	**Outside entity**
أطفال (children)	O	0.99	**Outside entity**
نتيجة (result)	O	1.00	**Outside entity**
السيول (the floods)	O	1.00	**Outside entity**
بتهالا (Bithala)	O	0.99	**Outside entity**
في (in)	O	0.67	**Outside entity** (Lower confidence)
غات (Ghat)	**B-LOC**	**0.78**	**Classified as "Location"**

2. Results of the Second Example (Sentence including names and places)

Table 6.4: Sentence including names and places

واللواء (and the Brigade)	B-ORG	0.68	Classified as "Organization" (Possibly a military unit name)
52 (52)	B-ORG	0.53	**Classified as "Organization" (Low confidence)**
بمحيط (in the vicinity of)	O	0.59	**Outside entity** (Low confidence)
منزل (house of)	O	0.52	**Outside entity** (Low confidence)
الدبيبة (Al-Dabaiba)	B-ORG	0.60	**Classified as "Organization" (Low confidence)**
في (in)	B-LOC	0.71	**Classified as "Location"**
حي (district)	B-LOC	0.81	**Classified as "Location"**
الاندلس (Al-Andalus)	B-LOC	0.87	**Classified as "Location"**

Table 6.5: Results of the Third Example (Sentence fragment including "Security")

Token	**Final Tag**	**Confidence**	**Notes**
الآليات (The mechanisms)	O	0.59	**Outside entity** (Low confidence)
التابعة (belonging)	O	0.64	**Outside entity** (Low confidence)
للأمن (to the Security)	**B-ORG**	**0.65**	**Classified as "Organization"**
العام (General)	O	0.51	**Outside entity** (Very low confidence)

اليوم	Today	O	0.74
خليفة	Khalifa	I-PERS	0.52
حفتر	Haftar	I-PERS	0.61
في	in	O	0.41
بنغازي	Benghazi	B-LOC	0.52
اليوم	Today	O	0.51
زار	visited	O	0.42
خليفة	Khalifa	I-PERS	0.65
حفتر	Haftar	I-PERS	0.74
سرت	Sirte	B-LOC	0.78
عبد	Abd	B-PERS	0.90
الحميد	Al-Hamid	B-PERS	0.91
الدبيبة	Al-Dabaiba	B-PERS	0.85
يصل	arrives	B-PERS	0.53
مطار	Airport	B-LOC	0.82
معتيقة	Mitiga	B-LOC	0.82
اليوم	Today	O	0.38
عبد	Abd	B-PERS	0.88

الحميد	Al-Hamid	B-PERS	0.87
الدبيبة	Al-Dabaiba	B-PERS	0.78
وصل	arrived	O	0.47
مطار	Airport	B-LOC	0.79
مصراتة	Misrata	B-LOC	0.77
زار	visited	O	N/A
خليفة حفتر	Khalifa Haftar	PERS	N/A
سرت	Sirte	LOC	N/A
زار	visited	O	0.37
حفتر	Haftar	I-PERS	0.63
سرت	Sirte	B-LOC	0.76
أختتم	concluded	O	N/A
اليوم	today	O	N/A
الأول	the first	O	N/A
من	of	O	N/A
مؤتمر ليبيا للتحول الطاقوي العادل	Libya Conference for Just Energy Transition	ORG	N/A
شركة	Company	O	0.41
الزاوية	Zawiya	I-ORG	0.52
لتكرير	for refining	I-ORG	0.60
النفط	Oil	I-ORG	0.55
شركة الزاوية لتكرير النفط	Zawiya Oil Refining Company	ORG	N/A
ذهب	went	O	N/A
أحمد	Ahmad	PERS	N/A

من	from	O	N/A
طرابلس	Tripoli	LOC	N/A
إلى	to	O	N/A
بنغازي	Benghazi	LOC	N/A
ذهب	went	O	N/A
خليفة	Khalifa	PERS	N/A
حفتر	Haftar	PERS	N/A
الى	to	O	N/A
مدينة	city of	O	N/A
سرت	Sirte	LOC	N/A
خليفة	Khalifa	B-PERS	0.51
حفتر	Haftar	I-PERS	0.55
طريق	road of	B-LOC	0.77
الشط	Al-Shat	I-LOC	0.74
منطقة	area of	B-LOC	0.68
الخلة	Al-Khalla	I-LOC	0.54
شي	something	O	0.69
جميل	beautiful	O	0.62
في	in	O	0.43
ليبيا	Libya	B-LOC	0.57
لما	when	O	0.87
تشوف	you see	O	0.77
انه	that	O	0.91
ف	so (in dialect)	O	0.90
اعمار	construction / reconstruction	O	0.91

ونشاطات	and activities	O	0.83
واحتفالات	and celebrations	O	0.79
ورفاهية	and welfare/luxury	O	0.85
اليوم	Today	O	0.94
بنمشي	I will go	O	0.60
لبنغازي	to Benghazi	B-LOC	0.65
بنقعد	I will stay	O	0.76
يوم	day	O	0.74
وبعدها	and after that	O	0.84
بنمشي	I will go	O	0.56
للبيضاء	to Al Bayda	I-ORG	0.52
مجلس	Council	B-ORG	0.67
النواب	Representatives	O	0.46
يعين	appoints	O	0.34
الشكري	Al-Shukri	B-PERS	0.63
بدل	instead	B-PERS	0.66
من	of	B-PERS	0.62
الصديق	Al-Siddiq	B-PERS	0.78
الكبير	Al-Kabeer	B-PERS	0.77
محافظا	Governor	B-ORG	0.52
لمصرف	for the Bank	B-ORG	0.58
ليبيا	Libya	O	0.48
المركزي	Central	I-ORG	0.62

6.4 Discussion

The findings of Arabic Named Entity Recognition (NER) task indicate some essential differences between the AraBERT and BiLSTM models that bring forward strength, weaknesses, and trade-offs of both models concerning the processing of Libyan Arabic text.

6.4.1 Overall Performance

It can be understood based on Table 6-1 that the BiLSTM architecture, when compared to AraBERT, demonstrates a slight advantage in both the overall Accuracy (0.94 vs. 0.92) as well as macro-average F1 -score (0.82 vs. 0.79). It implies that BiLSTM gives a more solid performance on all entity classes, especially in the areas with less frequent occurrence like LOC and ORG.

Conversely, AraBERT shows a marginally higher macro-average Recall (0.79), which is indicative of its ability to discover a higher number of true entities, in particular, the Person (PERS) classification. This feature suggests that AraBERT might be more effective on the tasks that require relating all the relevant entities rather than the high precision.

These results of the weighted F1 -score (0.93 in the case of BiLSTM and 0.92 in the case of AraBERT) prove that both of the models can handle the dominant type of class, which is O (Outside Entity); nevertheless, the slight advantage of BiLSTM in the weighted score denotes that the model has a higher generalizability to the overall distribution of entities in the dataset.

6.4.2 Entity-Specific Performance

Analysis of performance by entity category (Table 6.2) provides deeper insights:

1. **Organization (ORG):** These two models are both not good with ORG entities, though BiLSTM raises its F1 -score by 0.59 (AraBERT) to 0.68. This indicates that BiLSTM is in a better position to manage complex organizational boundaries, probably because of its sequence based structure that also considers the contextual dependencies in each direction.
2. **Location (LOC):** BiLSTM is better than AraBERT with an F1-score of 0.81 compared to 0.77 due to more precision (0.85). This implies that BiLSTM will be more precise in the process of determining locations, despite missing a few important tokens, as it correlates with its lower recall than AraBERT.

3. **Person (PERS):** AraBERT achieves a better balance between precision and recall (F1-score 0.86), largely due to its superior recall (0.83). Conversely, BiLSTM achieves very high precision (0.94) over recall (0.74), suggesting it tends to avoid false positives at the cost of missing some person entities.
4. **Outside Entities (O):** Both models achieve near-perfect performance (F1-score ≥ 0.96), as expected given the dominance of this class. This confirms that both models can reliably identify non-entity tokens.

6.4.3 Token-Level Classification Insights

To learn more about the model behaviors, the token-level analysis provides several qualitative observations that should be given a scholarly consideration:

- **Confidence variability:** In both the ORG and the PERS categories, there are relatively large numbers of tokens with a confidence rating between 0.50 and 0.70. This shows inherent vagueness in the predictions of these classes in the model especially when considering composite items such as military units and business names.
- **Error patterns:** In the BiLSTM architecture, error patterns exhibit a behavior that occurs naturally. It has a conservative bias, giving more importance to precision compared to recall. Contrastingly, AarBERT model is more inclusive being favorable to recall. This tension is critical when failure to include an entity is or is more expensive than the cost of false positive.
- **Context sensitivity:** BiLSTM has a benefit over its one-way counterpart because of context sensitivity that allows distinguishing between LOC and ORG tokens more precisely. A created edition named AraBERT exploits its pre-trained and default embedding contexts, which enhances the recall of PERS tokens to the cost of accuracy when dealing with the tokens of a limited or structurally complex nature.

6.4.4 Practical Implications

1. **Model selection:**

    - Use BiLSTM when you need to identify locations and organizations in a very strict manner, i.e. process news articles or government reports.
    - Use AarBERT when it is desired to find all persons mentioned e.g. during the analysis of social media posts or tracking social personalities.

2. **Hybrid approach**: A hybrid methodology, which deploys BiLSTM to the LOC and ORG token and uses AraBERT on the PERS token, would provide a better overall output and more balanced coverage of entities of each type.
3. **Confidence-based filtering:** Tokens with a lower confidence score than 0.65 can be subject to human intervention or a post-filtering rule, particularly with sensitive organizations, e.g. a government department, or even a military unit.

6.4.5 Discussion of Results and Comparison with Prior Studies

1. Superiority over Traditional and Early Deep Learning Approaches

- **Advantage over Feature-Based Approaches**

While Aldali [28] reported an F1-score of 94.21% for the "Organization" category using standardized Arabic text, the BiLSTM model in this study, applied to unstructured Libyan dialect data, achieved 70% F1 for Organization and 94% overall accuracy. This demonstrates the model's superior capability in capturing dialectal patterns and nuances that conventional feature-based methods fail to address effectively.

- **Advantage over Conventional BiLSTM Models on MSA**

The results of the BiLSTM model in this study (94% overall accuracy and 83% Macro F1) surpass the performance reported by El Bazi & Laachfoubi [29] on the ANERcorp dataset in Modern Standard Arabic (F1=90.6%). This finding highlights the model's robustness against morphological complexity and limited training data, underscoring its adaptability to colloquial dialects while maintaining high predictive performance.

2. Comparative Analysis of Transformer Model Performance

- **Performance versus AraBERT**

Within the same experimental context, BiLSTM achieved 94% overall accuracy, outperforming AraBERT's 92%. This result contrasts with typical expectations in the literature, where Transformer-based models often excel due to deep contextual representation. The superior performance of BiLSTM is attributed to the limited dataset size (4,000 tweets), where simpler architectures are less prone to overfitting and better exploit the available data.

- **Entity Category Comparison**

AraBERT exhibits a slight advantage in identifying Person (PERS) entities (F1=86%), reflecting its strong contextual understanding of named entities. In contrast, BiLSTM demonstrates a marked superiority in recognizing Organization (ORG) entities, outperforming AraBERT by 11%, which indicates its efficacy in modeling non-standardized organizational patterns commonly found in Libyan dialect social media text.

3. Qualitative Contribution to the Dialect Challenge (Libyan Dialect)

- **Advancement over Previous Libyan Models**

Prior research by Alfared & Alhammi [25] utilized a Rule-Based approach for NER in the Libyan dialect. The BiLSTM model presented in this study surpasses this earlier work, achieving over 94% accuracy and demonstrating the feasibility of applying deep learning techniques effectively to dialectal NER tasks.

- **Position Relative to Other Maghrebi Dialects**

Comparative evidence from studies on other Maghrebi dialects (e.g., Tunisian) indicates that NER in social media contexts remains challenging. The current study's Macro F1-score of 83%

underscores the persistence of these challenges and highlights the need for larger datasets and further research to fully capture the linguistic variability inherent in dialectal text.

6.4.6 Summary of Key Observations

- The BiLSTM model exhibits superior overall performance, particularly in the recognition of Location (LOC) and Organization (ORG) entities.
- AraBERT demonstrates enhanced recall for Person (PERS) entities, reflecting its strength in capturing contextual cues.
- Both models maintain excellent performance in detecting non-entity tokens, confirming their reliability for general token classification.
- Employing hybrid strategies or confidence-threshold filtering may further enhance model effectiveness, especially for dialects with sparse or structurally complex entities

Chapter Seven:
Conclusion and Future Works

7.1 Conclusion

In this study, we compared the two types of neural networks, AraBERT which is a Transformer style language model, fined-tuned on Arabic, and BiLSTM which is a bidirectional long short-term memory network, applied to the Named Entity Recognition (NER) task on Libyan dialect texts. Our major interest in the investigation was which of the models handles the unique linguistic issues associated with low-resource and dialect-rich data.

The analytical results and empirical evidence that follow (in Chapter 6) lead to a number of valid conclusions:

1. **Overall Superiority of BiLSTM in the Dialect Context:** The BiLSTM model also became the most self-balanced system of all the entity classes, with a Macro-Averaged F1-Score of 0.82, and AraBERT has a score of 0.79. Although the architecture of the BiLSTM is relatively less intricate, the sequential, bi-directional processing seems more convenient to address the morphological deviation and non-standard lexical representations, which are characteristic of the Libyan dialect.
2. **Challenges in the Organization (ORG) Category:** The worst performance was always observed with the ORG label, where F1 -score maximized at only 0.68. This deficiency probably can be attributed to ambiguities that occur in the process of drawing the line between formal Modern Standard Arabic entities and their dialects, enhanced by the irregular boundaries of such entities and spelling differences.
3. **AraBERT's Strength in Person Entity Recall:** AraBERT showed a marked Person (PERS) class level of recall, with a score of 0.83. This result indicates that learned contextual representation of large Arabic corpora will be useful even in a dialectal context, which demonstrates the effectiveness of AraBERT in capturing the semantic relationships and identifying personal names in a relatively low amount of dialect-specific training data.
4. **Token-Level Insights and Model Trade-Offs:**

- BiLSTM model had better precision especially on LOC and ORG whereas it had slightly lower recall on Person entity. In contrast, predictions in AraBERT were more broad-based and focused on the recall and not the precision.

- These complementary behaviors shed some light on the classical precision-recall trade-off and lead to possible advantages of hybrid or ensemble methods that have the ability to trade off coverage and accuracy.

Overall, the two models have reached significant conclusions in the limitations of a resource-restricted and dialect-filled environment. Although BiLSTM is more suitable in the case of balanced, general-purpose NER tasks, AraBERT can probably be a preferred model in the case where recall is important, in particular in a situation where personal names need to be extracted correctly.

7.2 Future Recommendations

Based on the existing evidence, it is possible to investigate some directions to enhance NER to the dialect of the Libyan language:

1. **Building and Standardizing Libyan Resources:** The creation of large, annotated collections of NER datasets in the Libyan dialect is a very important task. These organized resources would have great benefits in model training, evaluation and generalization and consequently alleviate the limitation on scarce data.
2. **Dialect-Specific Pre-training:** Continuous pre-training a AraBERT on large amounts of unlabeled texts in Libyan dialect predictor suggests that the model will be fined to the dialect-specific linguistic patterns, resulting in a higher downstream NER performance.
3. **Studying Code-Mixing Phenomena:** Since the Libyan dialect itself is often exposed to foreign terms and is often a borrowed word or term even in the name of an organization, an in depth study of the processes of code-mixing can significantly positively influence the recognition of the entity where there is a multilingual portion of the language.
4. **Incorporating Conditional Random Fields (CRF):** In addition to BiLSTM and AraBERT designs, including a CRF layer would provide logical and sequential consistency in entity labelling and decrease annotation mistakes in problematic or ambiguous contexts.
5. **Hybrid or Ensemble Models:** A combinational approach whereby BiLSTM is used in assessing LOC/ORG, and AraBERT is used in evaluating PERS would give the best overall results, especially in areas where high reliability of all entities is vital.

References:

1. Nadeau, D., & Sekine, S. (2007). *A survey of named entity recognition and classification*. Linguisticae Investigationes, 30(1), 3–26.
2. Yadav, V., & Bethard, S. (2019). *A survey on recent advances in named entity recognition from deep learning models*. In Proceedings of IJCAI.
3. Owens, J. (2006). *A linguistic history of Arabic*. Oxford University Press.
4. Habash, N., Rambow, O., & Kiraz, G. (2012). Morphological analysis and generation for Arabic dialects. In *Proceedings of COLING*.
5. Alotaibi, S., & Bouali, S. (2021). The lack of annotated datasets and linguistic resources for Arabic dialects. *Journal of Arabic Linguistics*, 34(2), 145–158.
6. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.
7. Qwaider, W., Talib, M. A., & Sembok, T. M. (2018). Arabic named entity recognition using deep learning: A review. *Journal of Computer Science*, 14(6), 876–887.
8. Ghosh, M., & Thirugnanam, A. (2021). Introduction to artificial intelligence. In *Artificial intelligence for information management: A healthcare perspective* (pp. 23–44). Springer Singapore.
9. Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., … Zhang, J. D. (2020). An introduction to machine learning. *Clinical Pharmacology & Therapeutics*, *107*(4), 871–885.
10. Alpaydin, E. (2014). *Introduction to machine learning*. MIT Press.
11. Bishop, C. M. (2007). Pattern recognition and machine learning. *Journal of Electronic Imaging*, 16(4), 049901.
12. Ruder, S., Peters, M. E., Swayamdipta, S., & Wolf, T. (2019). Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials* (pp. 15–18).
13. Kingma, D. P., Rezende, D. J., Mohamed, S., & Welling, M. (2014). Semi-supervised learning with deep generative models. *Advances in Neural Information Processing Systems*, 27.

14. Gaweda, A. E., Muezzinoglu, M. K., Aronoff, G. R., Jacobs, A. A., Zurada, J. M., & Brier, M. E. (2005). Individualization of pharmacological anemia management using reinforcement learning. *Neural Networks, 18*(5–6), 826–834.
15. Bozinovski, S. (1982). A self-learning system using secondary reinforcement. *Cybernetics and Systems Research*, 397–402.
16. Samuel, L. (2000). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44(1–2), 207–219.
17. Niu, T., & Bansal, M. (2018). Adversarial over-sensitivity and over-stability strategies for dialogue models. In *Proceedings of CoNLL*, 486–496.
18. Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828.
19. Tillmann, M. (2015). On the computational intractability of exact and approximate dictionary learning. *IEEE Signal Processing Letters*, 22(1), 45–49.
20. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
21. Guirguis, M. (2023). *Named entity recognition from biomedical text*. AUC Knowledge Fountain.
22. Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv:1508.01991*.
23. Al-Yousef, G. (2020). *Arabic natural language processing using deep learning* (Master's thesis). Khalifa University, UAE.
24. Younes, J., Achour, H., Souissi, E., & Ferchichi, A. (2020). A deep learning approach for Romanized Tunisian dialect identification. *IAJIT*, 17(6), 2020–2030.
25. Alfared, R. A., & Alhammi, H. (2018). Libyan dialect named entity recognition on rule-based approach. In *Proceedings of LICEET*, Tripoli.
26. Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., & Roth, R. (2014). *MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic*. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*.
27. Benajiba, Y., Rosso, P., & Zitouni, I. (2009). Arabic named entity recognition: A feature-based approach. In *ACL-IJCNLP Workshop*.

28. Aldali, N. M. (2018). A combination method of linguistic features and machine learning techniques for identifying Arabic named entities. *ICETR*, Elmergib University.
29. El Bazi, M., & Laachfoubi, N. (2019). Arabic named entity recognition using deep learning approach. *IJECE,* 9(3), 2025–2032.
30. Alzaidi, B. S., Abushark, Y., & Khan, A. I. (2022). Arabic Location Named Entity Recognition for Tweets using a Deep Learning Approach. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 13(12), 76–83.
31. Alsaaran, N., & Alrabiah, M. (2021). Arabic named entity recognition: A BERT-BGRU approach. *Computers, Materials & Continua*, 67(1), 123–135.
32. Al-Qurishi, M., & Souissi, R. (2022). Arabic named entity recognition using transformer-based CRF model. *Elm Research Department*, Riyadh.
33. Jarrar, M., Khalilia, M., & Ghanem, S. (2022). Wojood: Nested Arabic named entity corpus and recognition using BERT. *arXiv:2205.09651*.
34. Wadhawan, A. (2021). Dialect identification in nuanced Arabic tweets using Farasa segmentation and AraBERT. *EACL-WANLP*.
35. Talafha, B., Ali, M., Za'ter, M. E., Seelawi, H., Tuffaha, I., Samir, M., Farhan, W., & Al-Natsheh, H. T. (2020). *Multi-Dialect Arabic BERT for Country-Level Dialect Identification*. Mawdoo3 Ltd, Amman, Jordan.
36. Harrat, S., Meftouh, K., & Smaili, K. (2018). Maghrebi Arabic dialect processing: An overview. *JISGA*, 1(1).
37. Ramadan, A., & Hanan, A.,: (2024) , "Sentiment Analysis Of Libyan Tweets Using Machine Learning Algorithms", Malaysian Journal of Industrial Technology (MJIT), Volume 8, No.1, Mar 2024 eISSN: 2637-1081.
38. CHRIF, M. E. M. E. A., Seyed, C., Mahmoud, C. M., El Benany, M. M., Mint Mohamed-Saleck, F., Mohamed Saleck, M., El Beqqali, O., & Nanne, M. F. (2023). *Investigate the Impact of Stemming on Mauritanian Dialect Classification using Machine Learning Techniques*. International Journal of Advanced Computer Science and Applications (IJACSA), 14(10).
39. Elkhbir, N. (2024). *Information extraction for arabic and its dialects* (Doctoral dissertation, Université Paris-Nord-Paris XIII).
40. Konooz. (2025). Multi-domain multi-dialect corpus for named entity recognition. *arXiv*.

41. Singh, K., Sharma, R., & Gupta, M. (2025). NLP in artificial intelligence: Challenges and emerging trends. *Journal of AI Research and Applications*, 5(2).
42. Alayba, M. (2025). Arabic NLP: A comprehensive review. *Computers*, 14(11), 497.
43. Al Deen, M. M. S., Pielka, M., Hees, J., Abdou, B. S., & Sifa, R. (2021). *Improving Natural Language Inference in Arabic using Transformer Models and Linguistically Informed Pre-Training*. Fraunhofer IAIS & Hochschule Bonn-Rhein-Sieg, Germany.
44. Saadiyeh, O., Ramadan, A., Hajjar, M., & Bernard, G. (2025). *A comparative study of Arabic syntactic analyzers*. Frontiers in Artificial Intelligence. https://doi.org/10.3389/frai.2025.1638743
45. Aldawsari, M., Kolhar, M., & Dawood, O. S. (2023). Within-document Arabic event coreference. *Applied Sciences*, 13(19).
46. Shiva, S., El Dosuky, M., & Kamel, S. (2024). NLP and NLU techniques for intelligent search. *IJCA*, 186(11).
47. Qu, X., Gu, Y., Xia, Q., Li, Z., Wang, Z., & Huai, B. (2023). *A Survey on Arabic Named Entity Recognition: Past, Recent Advances, and Future Trends.* IEEE Transactions on Knowledge and Data Engineering, 36(3), 943–959.
48. Al-Kabi, S., et al. (2023). Arabic spelling and grammar error detection and correction: A review. *Arabian Journal for Science and Engineering*, 48.
49. Al-Smadi, M., et al. (2023). Arabic text classification using deep learning: A comprehensive survey. *JKSUCIS*, 35(7).
50. Keraghel, I., Morbieu, S., & Nadif, M. (2023). *Recent advances in named entity recognition: A comprehensive survey and comparative study*. Centre Borelli, Université Paris Cité.
51. Alex, B., Haddow, B., & Grover, C. (2007). Recognising nested named entities in biomedical text. In *BioNLP Workshop*.
52. Farrugia, K. (2022). *Multilingual transformer models for Maltese named entity recognition* (Master's thesis). Uppsala University.
53. Panchendrarajan, R., & Amaresan, A. (2018). Bidirectional LSTM-CRF for named entity recognition. School of Computing, National University of Singapore.
54. Sherstinsky, A**.** (2020). *Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network*. Physica D: Nonlinear Phenomena, 404**.**

55. Samih, Y. (2017). *Dialectal Arabic processing using deep learning* (Doctoral dissertation). Heinrich-Heine-Universität Düsseldorf.
56. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*. In I. Guyon et al. (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc.
57. Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools* (pp. 9–15).
58. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of deep bidirectional transformers for language understanding* (arXiv:1810.04805). arXiv