ROYAUME DU MAROC

UNIVERSITE MOHAMMED PREMIER

Ecole Nationale des Sciences Appliquées

Oujda - Maroc

Mémoire de Fin d'Année

Filière : Ingénierie du Data Science et Cloud Computing

Explainable AI for Arabic Web Pages Classification

Réalisé par :

SAIHI AMINE
EL MARCHOUM AYOUB
BELHAID NASSROU-EDDINE

Jury d'examen :

M. Abdelmounaim Kerkri
M. Mohamed Amine Madani

Encadrants :
M. Abdelmounaim Kerkri

Année universitaire
2023/2024

Acknowledgement

We would like to express our sincere gratitude to all the individuals who have contributed to the realization of this project. In particular, we would like to thank:

- Mr. Abdelmounaim KERKRI, our supervising professor, for his valuable guidance and support throughout this project.
- We extend our heartfelt thanks to the jury member, Mr. Mohammed Amine Madani, for dedicating their time and expertise to evaluate our End-of-Year Project. Their presence and invaluable feedback greatly contributed to the improvement of our work.
- Mr. Aissaoui Mohammed, our head of department, for his support and encouragement.
- Mr. Lehbil, the school director, for his support towards our major.
- Our parents and families, for their love, unconditional support, and encouragement throughout our studies.

Their support and encouragement have been essential to successfully completing this work. We are grateful for their contribution.

Remerciements

Nous tenons à exprimer notre sincère gratitude à toutes les personnes qui ont contribué à la réalisation de ce projet. En particulier, nous tenons à remercier

- M. Abdelmounaim KERKRI, notre professeur superviseur, pour ses précieux conseils et son soutien tout au long de ce projet. pour ses précieux conseils et son soutien tout au long de ce projet.
- Nous remercions chaleureusement les membres du jury, M. Mohammed Amine Madani, pour avoir consacré son temps et son expertise à l'évaluation de notre projet de fin d'année. Leur présence et leurs précieux commentaires ont grandement contribué à la réussite de notre projet. Leur présence et leurs précieux commentaires ont grandement contribué à l'amélioration de notre travail. travail.
- M. Aissaoui Mohammed, notre chef de filière, pour son soutien et ses encouragements. pour son soutien et ses encouragements.
- M. Lehbil, le directeur de l'école, pour son soutien à l'égard de notre majeure.
- Nos parents et familles, pour leur amour, leur soutien inconditionnel et leurs encouragements tout au long de nos études. tout au long de nos études.

Leur soutien et leurs encouragements ont été essentiels pour mener à bien ce travail. travail. Nous leur sommes reconnaissants pour leur contribution.

Abstract

Our project, titled **"Explainable AI for Arabic Web Pages Classification"**, applies **natural language processing (NLP)** techniques to classify Arabic text dataset. We explored models that are variants of **Artificial Neural Networks (ANN)**, including **Long Short-Term Memory (LSTM)** networks, and **Bidirectional Encoder Representations from Transformers (BERT)**. Using various libraries and frameworks, we compared the performance of each model to identify the most effective one.

Our experiments showed the following validation accuracies: **Arabert** fine-tuned achieved 96.23%, **LSTM** and **BILSTM** with **USE** embedding achieved 75.23%, **KNN** with USE embedding achieved 69.00%, **LSTM-BILSTM** with **Fasttext** embedding achieved 96.75%, and KNN with Fasttext embedding achieved 95.70%.

We also integrated **Explainable Artificial Intelligence (XAI)** techniques, including **LIME**, **SHAP**, and **Integrated Gradients**, to enhance the transparency and interpretability of our models. These XAI methods allowed us to provide detailed explanations for our model's predictions, fostering trust and enabling more informed decision-making. Through our efforts, we aimed to contribute to the growing field of explainable AI, especially for Arabic web page classification.

Résumé

Notre projet, intitulé **"Explainable AI for Arabic Web Pages Classification"**, applique des techniques de **traitement du langage naturel (NLP)** la classification de text Arabe. Nous avons exploré des modèles qui sont des variants des **réseaux de neurones artificiels (ANN)**, y compris les réseaux **Long Short-Term Memory (LSTM)** et les **Bidirectional Encoder Representations from Transformers (BERT)**. En utilisant diverses bibliothèques et frameworks, nous avons comparé les performances de chaque modèle afin d'identifier le plus efficace.

Nos expériences ont montré les précisions de validation suivantes : **Arabert** finetuned a atteint 96.23%, **LSTM** et **BILSTM** avec l'embedding **USE** ont atteint 75.23%, **KNN** avec l'embedding USE a atteint 69.00%, **LSTM-BILSTM** avec l'embedding **Fasttext** a atteint 96.75%, et KNN avec l'embedding Fasttext a atteint 95.70%.

Nous avons également intégré des techniques d'**Intelligence Artificielle Explicable (XAI)**, y compris **LIME**, **SHAP** et **Integrated Gradients**, pour améliorer la transparence et l'interprétabilité de nos modèles. Ces méthodes XAI nous ont permis de fournir des explications détaillées pour les prédictions de notre modèle, favorisant ainsi la confiance et permettant une prise de décision plus éclairée. Grâce à nos efforts, nous avons cherché à contribuer au domaine croissant de l'IA explicable, en particulier pour la classification des pages web arabes.

Report Structure

This report is organized into five main chapters, each addressing a critical component of the project.

- **Chapter 1: General Introduction** - This chapter provides an overarching introduction to the project. It sets the context, defines the research problem, outlines the objectives, and discusses the contributions of the study.
- **Chapter 2: State of the Art** - In this chapter, we review the current advancements in the field of Arabic NLP. It covers various aspects of text classification, Explainable AI (XAI), and the historical and modern applications of NLP. Key challenges, approaches, and recent advancements in XAI, especially related to Arabic NLP, are also discussed.
- **Chapter 3: Methodology** - This chapter details the methodologies used throughout the project. It covers data collection, tokenization, embedding techniques, modeling approaches including transfer learning and fine-tuning, and the explainability methods employed, such as LIME, SHAP, and Integrated Gradients. The rationale behind choosing specific explainability techniques is also explained.
- **Chapter 4: Implementation and Results** - Here, we present the implementation details and the results of the models. This chapter includes performance metrics and interpretability of various models, such as BERT, LSTM, and KNN. It also discusses the impact of XAI, challenges faced in Arabic NLP, and concludes with insights and future perspectives.
- **Chapter 5: Conclusion and Future Work** - The final chapter summarizes the key findings of the report, discusses the implications of the research, and suggests directions for future work.

The report also includes appendices and references for additional context and resources.

Contents

List of Figures

Chapter 1

General Intoduction

1.1 Introduction

In today's digital era, the vast amounts of textual data generated daily across diverse languages present both challenges and opportunities. Arabic, with its rich linguistic heritage and cultural significance, stands out as a particularly intricate language for text processing. As organizations seek to extract meaningful insights from Arabic texts, the fusion of Natural Language Processing (NLP) techniques with machine learning and deep learning models offers a promising avenue for analysis and comprehension.

The application of machine learning and deep learning models to Arabic text classification has revolutionized how we navigate and understand vast corpora of textual data. These models, fueled by large-scale datasets and sophisticated algorithms, enable automatic categorization, sentiment analysis, and information extraction from Arabic texts with unprecedented accuracy and efficiency. However, the inherent complexity of Arabic, characterized by its intricate morphology and syntactical nuances, poses unique challenges for text classification tasks.

Moreover, as AI systems become increasingly integrated into critical decision-making processes, the need for transparency and interpretability has become paramount. Enter Explainable AI – a burgeoning field that seeks to demystify the inner workings of machine learning models, making their decisions comprehensible to human users. Incorporating Explainable AI principles into Arabic text classification not only enhances the trustworthiness of AI-driven insights but also fosters deeper understanding and collaboration between machines and humans.

In this report, we delve into the realm of Arabic text classification using NLP techniques, machine learning, and deep learning models, augmented by Explainable AI methodologies. We explore the intricacies of Arabic language processing, dissect the challenges

inherent in text classification, and elucidate the transformative potential of Explainable AI in enhancing the transparency and interpretability of AI-driven text classification systems. Through empirical analysis, practical applications, and insightful discussions, we aim to shed light on the evolving landscape of Arabic text classification and pave the way for future advancements in this exciting field.

1.2 Context

In the current landscape of digital expansion and the proliferation of textual data from various sources such as social media, news articles, and corporate communications, the need to efficiently process and understand this data is becoming increasingly urgent. Particularly for organizations like Al Jazeera, operating in a dynamic and multilingual media environment, the ability to quickly and accurately classify and analyze textual content holds strategic importance.

Within this context, this study focuses on the application of various text classification techniques and models to address the specific needs of Al Jazeera in content management and analysis. By leveraging advanced natural language processing (NLP) approaches, this study aims to develop robust and accurate text classification systems capable of efficiently handling streams of textual data from diverse sources.

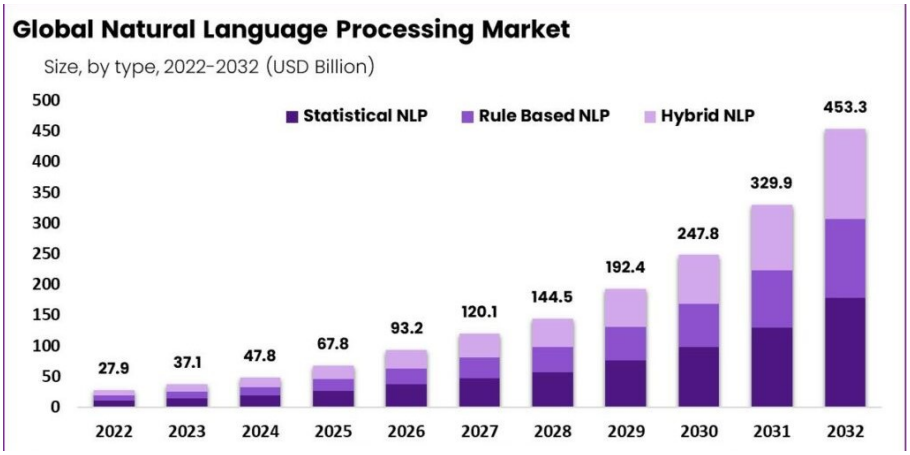


Figure 1.1: Explosive Growth: Global NLP Market Trends

Additionally, this study will address the concept of explainable artificial intelligence (AI) to ensure that the decisions made by the text classification models are transparent, understandable, and justifiable. This approach will not only enhance confidence in the classification outcomes but also provide valuable insights into how these models operate and make decisions, which is crucial in a context where reliability and transparency are top priorities.

Automatic classification of web pages is crucial for applications such as content filtering, personalized search, and monitoring inappropriate content, ensuring optimal results through the use of powerful NLP models and explainability techniques

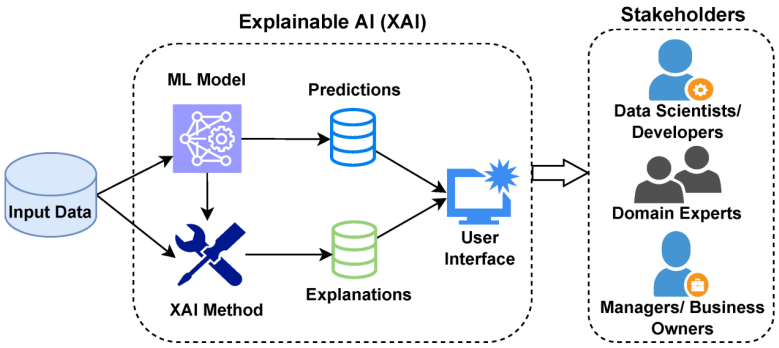


Figure 1.2: Explainable AI serves to provide comprehensive reasons behind model predictions to non-specialists

1.3 Problematic

Despite significant advancements in Natural Language Processing (NLP), Arabic NLP remains underdeveloped compared to its counterparts in other languages, primarily due to the scarcity of high-quality Arabic datasets and the complexity of the Arabic language itself. This research gap poses challenges for developing robust and accurate Arabic text classification models. Additionally, the application of Explainable AI (XAI) in Arabic NLP is relatively nascent. While state-of-the-art models such as transformers have shown promise, their opaque nature raises concerns about transparency and trustworthiness. Therefore, there is a critical need for integrating XAI principles into Arabic NLP models to enhance their interpretability and reliability. This research aims to address these gaps by developing advanced Arabic text classification models and incorporating XAI techniques to ensure the model's decisions are transparent and understandable.

1.4 Objectives

The objective of this report is to provide a comprehensive analysis of Arabic text classification techniques. This includes an in-depth examination of both traditional machine learning methods and cutting-edge deep learning models specifically tailored for processing the Arabic language.

Integration of Explainable AI Principles: Investigate the incorporation of Explainable AI principles into Arabic text classification models to improve transparency, interpretability, and trustworthiness of the classification results.

andscape of digital expansion and the proliferation

1.5 Contributions

This report makes several contributions to the field of Arabic text classification and Explainable AI. Firstly, it provides a comprehensive synthesis of existing text classification techniques tailored for Arabic language processing, offering insights into their strengths, weaknesses, and specific applications. Additionally, it offers a critical analysis of current methods, identifying areas for improvement and guiding future research endeavors. Furthermore, this report includes the implementation and evaluation of classification models integrated with Explainable AI principles, demonstrating their effectiveness through empirical assessments and real-world applications. Through detailed case studies and practical examples, it illustrates the relevance and potential impact of Arabic text classification techniques in various domains, ranging from social media analysis to information retrieval. Lastly, by offering practical recommendations for both practitioners and researchers, this report serves as a valuable resource for advancing the state-of-the-art in Arabic text analysis and fostering interdisciplinary collaborations in the field.

Chapter 2

State of the Art

2.1 Introduction

In this chapter, we will explore the current developments in Natural Language Processing (NLP) relevant to our project, with a focus on arabic text classification and Explainable AI (XAI).

2.2 Text classification

Text classification is a fundamental task in natural language processing (NLP) that involves assigning one or more predefined labels to a text document. This task has witnessed significant growth in recent years, driven by the explosion of textual data and advancements in machine learning techniques. The early work on text classification dates back to the 1960s with the emergence of question-answering information systems. Statistical approaches, such as naive Bayes classification, were widely used in the 1970s and 1980s. The 1990s saw the rise of machine learning algorithms based on decision trees and artificial neural networks, which led to improvements in text classification performance. Text classification relies on two main steps:

- ***Text Representation:*** The text is converted into a numerical representation that the classification algorithm can understand. This can be done using techniques such as bag-of-words, n-grams, or word embeddings.
- ***Learning and Classification:*** A machine learning algorithm is trained on a dataset of labeled texts. The algorithm learns to identify patterns in the data and use them to classify new texts.
  Text classification has a wide range of applications, including: *Email Routing:* Classifying emails into folders such as "spam," "inbox," or "promotions." *Sentiment*

Analysis: Determining the sentiment of a text, whether positive, negative, or neutral. *Document Categorization:* Classifying documents into predefined categories, such as "news," "sports," or "finance."...

Figure 2.1: Text classification illustration

2.3 Explainable AI (XAI)

Explainable AI (XAI) refers to a set of processes and methods that allow human users to understand and trust the outcomes and predictions made by artificial intelligence models. The goal of XAI is to make AI systems decision-making processes transparent, interpretable, and comprehensible to humans, ensuring that the AI's actions can be understood, explained, and justified.

XAI is a rapidly growing field that aims to enhance the transparency and interpretability of AI systems. As AI models become increasingly complex and integrated into critical decision-making processes, understanding their reasoning and decision-making processes is crucial for building trust and ensuring ethical and responsible AI development.

2.3.1 Key Challenges in XAI

Developing effective XAI techniques presents several challenges:

- **Black-Box Models:** Many AI models, particularly deep neural networks, are inherently complex and opaque, making it difficult to understand how they arrive at their decisions.
- **Multiple Explanations:** Different users may require different levels of explanation and may prioritize different aspects of the model's reasoning.
- **Context-Dependent Explanations:** Explanations should be contextualized to the specific input data and decision being made.

- **Human Understanding:** Explanations should be presented in a way that is understandable to humans, considering their domain knowledge and cognitive biases.

2.3.2 Approaches to XAI

To address these challenges, various XAI approaches have been proposed, categorized into three main groups:

- **Model-Agnostic Methods:** These methods work independently of the specific AI model, analyzing the input data and output predictions to generate explanations.
- **Model-Specific Methods:** These methods leverage knowledge of the internal structure and workings of the AI model to provide more detailed and interpretable explanations.
- **Human-in-the-Loop Methods:** These methods involve human experts in the loop, iteratively refining explanations based on their feedback and domain knowledge.

2.3.3 Applications of XAI

XAI has a wide range of applications across various domains, including:

- **Healthcare:** Explaining AI-driven medical diagnoses and treatment recommendations to improve patient trust and informed decision-making.
- **Finance:** Providing insights into AI-based risk assessments and fraud detection to enhance transparency and accountability.
- **Criminal Justice:** Explaining AI-powered recidivism risk predictions to support fairer sentencing and parole decisions.

2.3.4 Recent Advancements in XAI

The field of XAI is witnessing rapid advancements, with new techniques and tools emerging continuously:

- **Local Interpretable Model-Agnostic Explanations (LIME):** A model-agnostic method that generates explanations by approximating the model's behavior around a specific prediction.
- **Gradient-based Methods:** These methods explain model decisions by analyzing the gradients of the model's output with respect to its input features.

- **Attention Mechanisms:** These methods provide insights into which parts of the input data the model is focusing on when making a decision.
- **Counterfactual Explanations:** These methods generate alternative scenarios that would have led to a different model prediction, helping to understand the factors influencing the decision.

2.3.5 Related work in Arabic XAI

Despite the rise of large language models (LLMs) and generative models, the quality of Arabic data remains inferior to that of other languages. This is primarily due to the scarcity of comprehensive Arabic datasets. Regarding Arabic Explainable AI (XAI), we found a noteworthy paper [1] that applies XAI techniques to Sentiment Analysis on Arabic text data, using the LIME method.

2.3.6 Future Directions in XAI

The future of XAI lies in developing more robust, reliable, and human-centric XAI techniques that can effectively address the challenges of complex AI models and diverse user needs. Key areas of focus include:

- **Explainability by Design:** Integrating explainability principles into the design and development of AI models from the outset.
- **Personalized Explanations:** Tailoring explanations to individual users based on their background, knowledge, and preferences.
- **Interactive Explanations:** Enabling interactive exploration of explanations to facilitate deeper understanding and trust.
- **Formal Methods for XAI:** Developing formal frameworks and tools to evaluate and assess the quality and trustworthiness of explanations.

[1] Arabic Sentiment Analysis with Noisy Deep Explainable Model

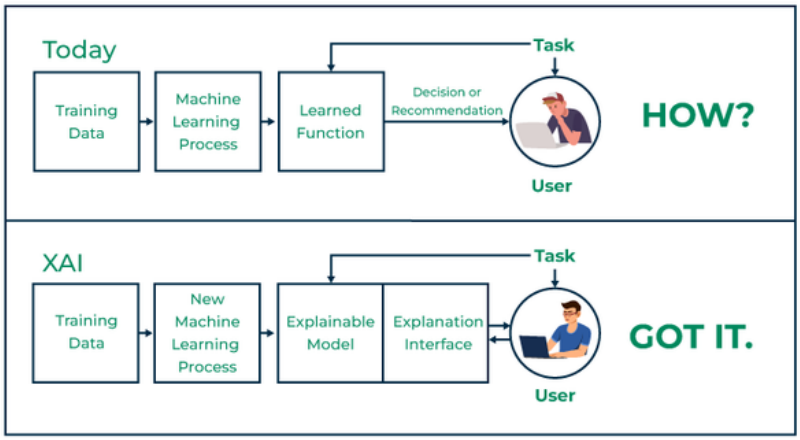


Figure 2.2: Explainable AI aims to understand the underlying mechanism that led to model prediction

2.4 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a rapidly evolving field that focuses on the interaction between computers and human language. It encompasses a wide range of tasks, from understanding and generating text to extracting information from natural language sources. NLP has become increasingly important in recent years due to the explosion of textual data and the growing demand for AI systems that can interact with humans in a natural way.

2.4.1 Historical Overview of NLP

The development of NLP has seen significant milestones over the decades:

- **1950s:** Rule-based systems were employed to translate texts from Russian to English during the Cold War.
- **1960s:** The emergence of the first conversational agents capable of rephrasing and paraphrasing sentences.
- **1980s:** Researchers combined rules and statistics to create more complex systems.
- **Since 2016:** Deep Learning revolutionized NLP with new model architectures.

2.4.2 State of the art in NLP

- **Simple Machine Learning Models:**
  - Initial techniques involved transforming text data into numerical vectors using methods like TF-IDF (Term Frequency-Inverse Document Frequency).

  - Traditional ML algorithms such as Naive Bayes and Support Vector Machines were employed for basic text classification tasks.
  - These methods were limited in their ability to capture the sequential nature and contextual nuances of language.

- **Recurrent Neural Networks (RNNs):**
  - RNNs introduced the capability to handle sequences of text and maintain context through hidden states.
  - Despite their advancements, RNNs struggled with long-term dependencies due to issues like vanishing gradients.

- **Long Short-Term Memory (LSTM) Networks:**
  - LSTMs were developed to address the limitations of RNNs, offering improved performance by better preserving information over longer sequences.
  - These networks became widely used for various NLP tasks requiring an understanding of sequential data.

- **CNN-LSTM Models:**
  - The combination of Convolutional Neural Networks (CNNs) with LSTMs led to models that leveraged the strengths of both architectures.
  - CNNs captured local patterns in the text, while LSTMs handled the sequential data, resulting in a comprehensive understanding of both local and global structures in language.

- **Transformer Models:**
  - The introduction of transformers marked a significant breakthrough, using self-attention mechanisms which was first introduced in the paper **Attention is all you need**[2] to process entire sequences simultaneously.
  - Transformers allowed for greater parallelization and more efficient training on large datasets.
  - Models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) set new benchmarks in various NLP tasks.
  - These models significantly improved the ability to understand and generate human language, pushing the boundaries of what is possible in NLP.

[2]Attention Is All You Need Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin

2.4.3 Modern NLP Applications

Today, NLP technology has achieved human-level performance in various tasks, thanks to advances in machine learning and deep learning techniques. Some common NLP applications include:

- **Sentiment Analysis:** Analyzing the sentiment expressed in text data to determine the writer's attitude or emotional tone.
- **Chatbots:** Developing conversational agents that can interact with users in a natural and intuitive manner.
- **Machine Translation:** Automatically translating text from one language to another with high accuracy.
- **Text Classification:** Categorizing text into predefined categories for tasks such as spam detection and topic classification.

2.5 Conclusion

In this chapter, we have reviewed the state-of-the-art in Arabic NLP. Additionally, we will outline the methodology used for training and explaining models specifically for Arabic data.

Chapter 3

Methodology

3.1 Introduction

In this chapter, we will detail the process and methodology undertaken for the project's implementation. Additionally, we will highlight key concepts that are critical to understanding the technical aspects of the implementation. These foundational elements are essential for comprehending the strategies and decisions made throughout the project.

3.2 Data Collection

Our dataset was meticulously curated from the Al Jazeera website, capturing a comprehensive array of articles classified into distinct categories. With a focus on precision, we ensured that our dataset exclusively drew from the Al Jazeera platform, guaranteeing a consistent and reliable source for our analysis. This curated collection encompasses a diverse range of topics, spanning politics, economics, health , thus facilitating robust insights and comprehensive evaluations in our classification project.

3.3 Tokenization and Embedding

In our case, tokenization and embedding play a crucial role in effectively transforming textual data into a numerical format understandable by machine learning models. Tokenization breaks down the text into discrete units, facilitating its processing and analysis. By segmenting the text into meaningful tokens such as words or subwords, we preserve the structure and meaning of the text, which is crucial for text classification tasks like ours. Simultaneously, the use of embeddings allows us to represent each token as a numerical vector that captures its semantic and contextual properties. This vector representation retains the relationships and nuances between words, enabling machine learning models

to understand and generalize from textual data. By combining tokenization and embedding, we obtain a rich and meaningful numerical representation of the text, essential for training accurate and robust classification models on our Arabic dataset.

3.3.1 Tokenization

Tokenization, a fundamental in natural language processing (NLP), traces its roots back to the early days of computer science and linguistics. Its origins lie in the need to break down human language into manageable units for computational analysis. From a mathematical standpoint, tokenization operates on the principle of segmentation, dividing a continuous sequence of characters or words into discrete tokens. This process serves as the foundation for various NLP tasks by providing a structured representation of text data, enabling algorithms to operate on individual units of meaning. Over time, tokenization techniques have evolved to accommodate the complexities of different languages and text types, contributing to advancements in machine learning and text processing applications.

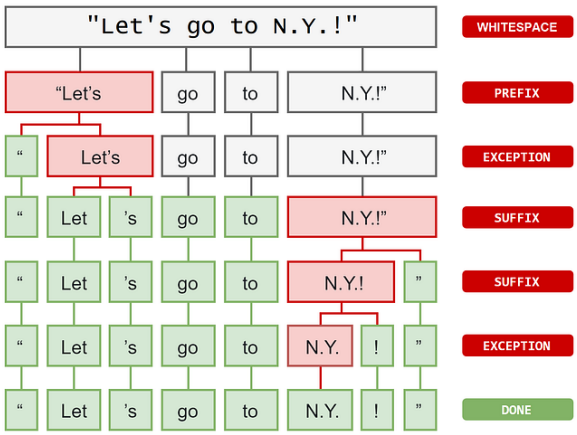


Figure 3.1: An illustration of tokenization

3.3.2 Embeddings

Word embeddings are an essential component in the field of Natural Language Processing (NLP). Mathematically, word embeddings represent a dimensionality reduction method that converts individual words into numerical vectors in a continuous vector space of lower dimensions. This transformation can be achieved using various models, for us we have used AraBERT, Universal Sentence Encoder, and FastText.

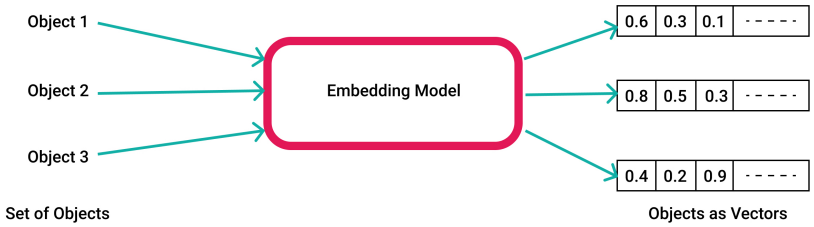


Figure 3.2: An embedding model serves to convert textual data to an algebric representation

AraBERT: Designed specifically for processing the Arabic language, AraBERT utilizes the BERT (Bidirectional Encoder Representations from Transformers) Introducted by google in the paper [1] architecture to create word embeddings. Each word in a text is initially represented as a one-hot vector, where each dimension corresponds to a unique word in a vocabulary. This one-hot vector is then transformed into a word embedding vector through a learned weight matrix, also known as an embedding matrix.

Universal Sentence Encoder (USE): USE focuses on producing embeddings for entire sentences rather than individual words. It captures the meaning of the sentence in a fixed-dimensional vector, which is useful for tasks involving sentence-level semantics. As described in the paper [2] Has two different encoders, a transformer based and a Deep average network (DAN)

FastText: FastText enhances word embeddings by considering subword information as mentioned in the paper[3]. It represents words as bags of character n-grams, allowing the model to generate embeddings for rare words or even out-of-vocabulary words by summing the vectors of their constituent n-grams.

The primary advantage of using these different types of embeddings lies in their ability to capture semantic and syntactic similarities between words or sentences. In the embedding vector space, similar words or sentences will be closer to each other. This proximity allows models to learn generalizations from training data and apply this knowledge to new, unseen data, thereby improving their performance on various NLP tasks.

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova

[2] Universal Sentence Encoder aGoogle Research Mountain View, CA bGoogle Research New York, NY cGoogle Cambridge, MA

[3] Enriching Word Vectors with Subword Information Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov

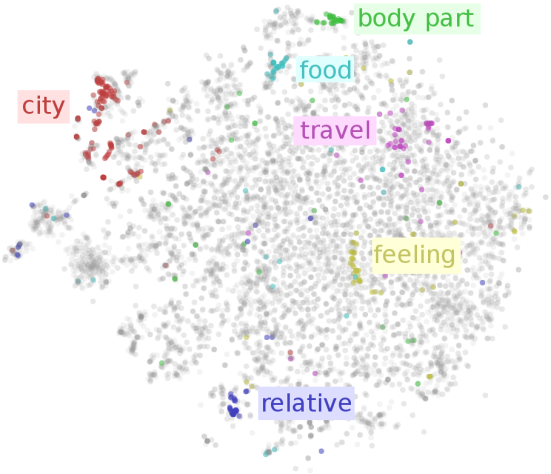


Figure 3.3: The idea behind embeddings in NLP is to represent Natural language contexts in n-dimensional space

3.4 Modelling techniques

3.4.1 Transfer Learning

Transfer learning is a machine learning technique where a model developed for a specific task is reused as the starting point for a model on a second task. This approach leverages the knowledge gained while solving one problem to address another related problem, which can significantly improve model performance and reduce training time.

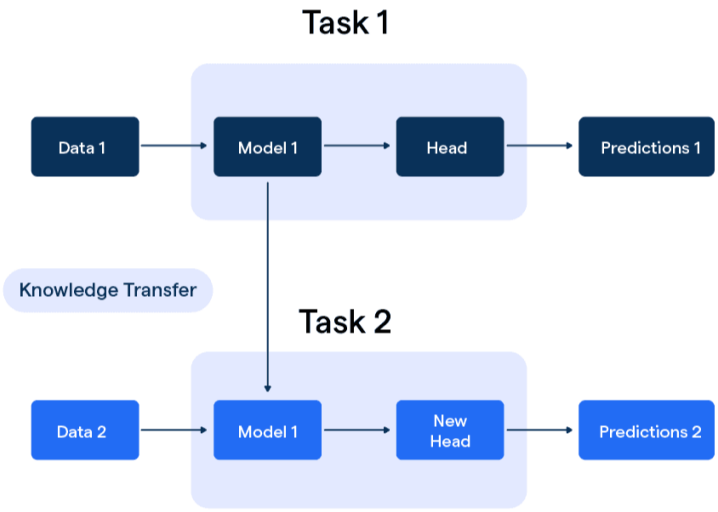


Figure 3.4: An iluustration of transfer learning, model learned parameters are transferred from task1 to task2

Traditional machine learning models are trained from scratch on specific datasets, requiring substantial data and computational resources. This approach often involves

long training times and may not perform well with limited data, necessitating retraining for new tasks even if they are similar.

In contrast, transfer learning leverages knowledge from large, diverse datasets to improve efficiency and performance on new, related tasks. It involves pretraining a model on a vast corpus to learn general features, then fine-tuning it on a smaller, task-specific dataset. This reduces training time and enhances performance, especially with limited data. In our project, we employ transfer learning using models like AraBERT, Universal Sentence Encoder (USE), and FastText, which are pretrained on extensive datasets, capturing rich semantic information that we adapt for our text classification tasks.

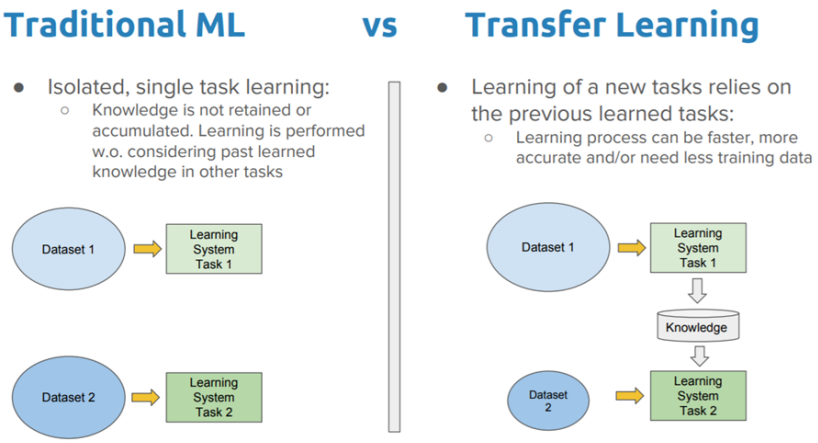


Figure 3.5: Traditional ML VS Transfer learning

3.4.2 Fine-tuning

Fine-tuning is a crucial step in transfer learning, involving the adjustment of a pre-trained model on a specific task with new, task-specific data. After pre-training on a large corpus to learn general representations, the model is refined by continuing its training on a smaller, targeted dataset. This step allows the model's general knowledge to be adapted to the nuances of the new task, thereby improving its performance. Fine-tuning significantly reduces the time and resources required compared to training a model from scratch and enables high-quality results even with limited task-specific data.

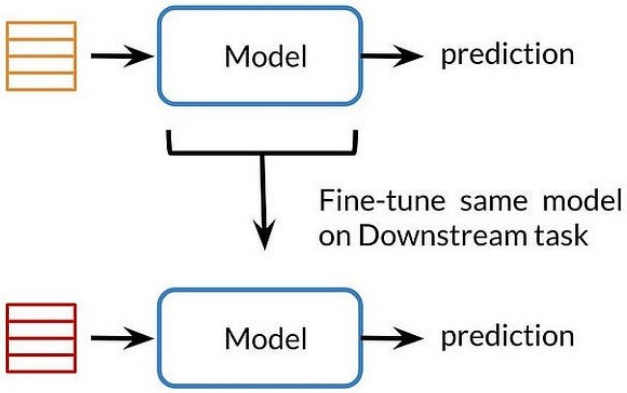


Figure 3.6: Fine-tuning aims to to tune a pre-trained model with learned parameters on a downstream task

3.5 Explainability

3.5.1 LIME

LIME (Local Interpretable Model-agnostic Explanations) is a technique used to explain the predictions of complex machine learning models by approximating them locally with interpretable models, such as linear regressions, that are weighted to emphasize instances similar to the input being explained.

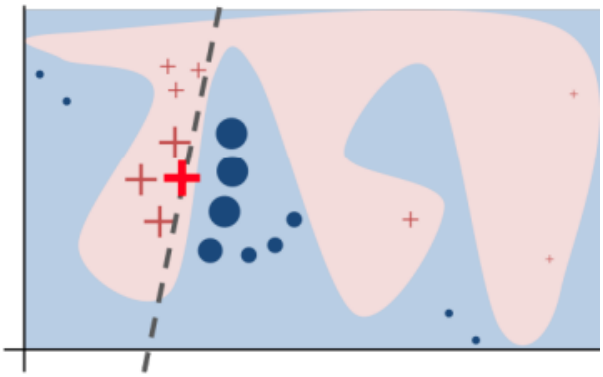


Figure 3.7: LIME's key idea is to perform a local approximation (instead of global) of the complex model around some perturbation instances noted π_x

The LIME objective function is designed to fit a simple interpretable model to approximate the predictions of a complex model in the local neighborhood of some generated instances. It is expressed in the paper [4] as follows:

$$\hat{g} = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{3.1}$$

[4]Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135-1144)

where:

- $\hat{g}$ is the interpretable model.
- G is the family of possible interpretable models (e.g., linear models).
- π_x is a proximity measure that defines the locality around x.
- $\mathcal{L}(f, g, \pi_x)$ is called the fidelity term, measuring how well the interpretable model g approximates the complex model f in the vicinity of the instance x.
- $\Omega(g)$ is the complexity term, penalizing the complexity of the interpretable model g to ensure it remains simple (low variance).

The fidelity term $\mathcal{L}(f, g, \pi_x)$ can be further specified as:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z \in Z} \pi_x(z) \left(f(z) - g(z)\right)^2 \tag{3.2}$$

where:

- Z is the set of perturbed samples around x.
- $\pi_x(z)$ is the weight of the sample z based on its proximity to x.
- $f(z)$ is the prediction of the complex model for the sample z.
- $g(z)$ is the prediction of the interpretable model for the sample z.

The final interpretable model is a sparse linear model, in the paper they have employed Lasso, for us we have tried both of lasso and ridge, as lasso tends to discriminate unimportant features whereas Ridge just reduce the complexity of the model, for textual data the features for the linear models are the words or tokens. So the feature importance are the coefficient of the sparse linear model, it's worth mentioning that the X input to the objective is weighted by $\pi_x(z)$

$$\hat{\beta} = \arg\min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} \pi_x(z_i)(y_i - X_i\beta)^2 + \alpha \|\beta\|_1 \right\} \tag{3.3}$$

the weights are computed by a similarity kernel, as we are in a natural language processing setting, cosine similarity is used, and the kernel is expressed as follows :

$$\pi_x(z_i) = \sqrt{\exp\left(-\frac{\cos(x, z_i)}{\text{kernel\_width}^2}\right)} \tag{3.4}$$

z_i is the perturbation generated from the input X, This can be done by randomly removing words (tokens) from the text. The idea is to create variations of the text to observe how the absence of certain words affects the prediction. these instances are then weighted by the similarity kernel in (3.4) with the original text input we are willing to explain, and then a linear model is fitted around the weighted instances and the coefficients will correspond to the feature importance for every token.

3.5.2 SHAP

SHAP which stands for SHapley Additive exPlanations, follows a similar approach, it compute a feature importance score called the **shapley value** for every token in the text, yet it does that in a different way, it computes every shap value directely in a single formula without an objective function.

- **Model** f: The complex machine learning model we want to explain.
- **Instance** x: The specific input for which we want to explain the prediction.
- **Feature set** F: The set of all features, $F = \{1, 2, \ldots, M\}$.
- **Shapley Value** ϕ_j: The contribution of feature j to the prediction for instance x.

SHAP assigns an importance value (Shapley value) to each feature, representing its contribution to the difference between the actual prediction and the average prediction.

- The Shapley value for a feature j is calculated as:

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{j\}}(x) - f_S(x) \right] \tag{3.5}$$

  where:

  - S is a subset of all features excluding j.
  - $f_S(x)$ is the prediction of the model using only the features in subset S.
  - $|S|$ is the number of features in subset S.
  - $|F|$ is the total number of features.

- For each feature j:

  a. Iterate over all possible subsets S of the feature set F excluding j.

  b. Compute the model prediction using the subset S and the subset $S \cup \{j\}$.

  c. Calculate the difference in predictions: $f_{S \cup \{j\}}(x) - f_S(x)$.

d. Weight this difference by the number of subsets of size $|S|$ and $|F| - |S| - 1$.

This method apparentely comes with a huge computational complexity, that is why there are plenty of variant of Shapley method including **Kernel SHAP** and **Tree SHAP**.

3.5.3 Integrated Gradient

- Integrated Gradients is one of the specific methods used to explain neural network models.
- It is an applicability method based on the calculation of gradients.
- **Function** : $F : \mathbb{R}^n \to [0, 1]$
  - Represents a neural network
- **Input and reference data** :
  - $x \in \mathbb{R}^n$: Input data
  - $x' \in \mathbb{R}^n$: Reference data
- **Segment** :
  - Connects x to x'
  - Computes the gradient at each point of this segment
- **IG Method** :
  - Consists of summing the gradients
- **IG formula in the i-th dimension** :

$$IG_i(x) = (x_i - x'_i) \int_0^1 \frac{dF(x' + \alpha(x - x'))}{dx_i} d\alpha \quad (1)$$

- **Advantage of IG** :
  - Satisfies the two axioms:
    - Sensitivity
    - Implementation Invariance

3.5.3.1 Sensitivity

- An explainability method satisfies sensitivity if, for any input and reference data that differ only at one feature, this method gives a non-zero attribution to this feature whenever the respective predictions associated with the input and the reference data are different.
- Let x be an input data point, x' a reference data point, and F a neural network function.
- $AF(F, x, x') = (a_1, a_2, \ldots, a_n)$ is an attribution method where a_i measures the contribution of x_i in the prediction $F(x)$.
- Then:

$$\forall x, x' \in \mathbb{R}^n, \forall i \in \{1, 2, \ldots, n\}; x - x' = u_i \text{ if } F(x) \neq F(x') \text{ then } a_i \neq 0$$

- Where $u_i = (0, \ldots, 0, k, 0, \ldots, 0)$ (with k being a non-zero real number at the i-th position)

3.5.3.2 Implementation Invariance Axiom

Two neural networks are said to be functionally equivalent if their outputs are the same for any input data, even if their architectures are different. An attribution method satisfies the implementation invariance axiom if the attributions are identical for two functionally equivalent networks.

- Let f and g be two neural networks. They are said to be functionally equivalent if for every input x, $f(x) = g(x)$.
- An attribution method ϕ satisfies the implementation invariance axiom if for every input x, the attributions given by ϕ for f and g are identical.
- Formally, this can be expressed as follows:
  - If $f(x) = g(x)$ for all x, then
  - $\phi^f(x) = \phi^g(x)$
  - where $\phi^f(x)$ and $\phi^g(x)$ represent the attributions calculated by the method ϕ for the networks f and g, respectively.

3.5.4 Choosing the Accurate Explainability Technique

In our project, we have focused on model-agnostic techniques which do not depend on the model architecture; they work regardless of the complex model used. There are other methods, such as attention visualization, that are used in models that use attention mechanisms, such as transformers and seq-to-seq architectures. Additionally, LIME and SHAP are used to explain token contributions, whereas we have used integrated gradients to explain embeddings contribution.

3.6 Conclusion

In this chapter, we have discussed the methodologies employed in our study and the key concepts associated with them. Specifically, we have explored three different embedding methods: FastText, Universal Sentence Encoder, and Arabert embeddings. Furthermore, we have fine-tuned Arabert embeddings for text classification tasks and trained LSTM, BILSTM, knn models using both FastText and Universal Sentence Encoder embeddings. In the upcoming chapter, we will delve into the results obtained from our experiments and conduct a comprehensive comparison of the performance of these different approaches.

Chapter 4

Implementation and Results

4.1 Introduction

In this chapter we will cover the results of our models training along with the explainability of each model.

4.2 Models performance and interpretation

This table summarizes the evaluation of each model and the corresponding performance.

Embedding Method	Model Architecture	Validation Accuracy
Arabert	Arabert fine-tuned	96.23
USE	LSTM, BILSTM	75.23
USE	KNN	69.00
Fastext	LSTM-BILSTM	96.75
Fastext	KNN	95.70

Table 4.1: Validation accuracy for the embeddings methods and model architectures

We have noticed that there is a huge difference between USE performance and other embeddings methods performances, although we have used hyper-parameter tuning for the LSTM models, we have used the python package **Optuna** that implements the **Tree-structured Parzen Estimator (TPE)** which is a Bayesian optimization algorithm that iteratively selects the next set of hyperparameters to evaluate, yet we have noticed that Fasttext just performs well over USE embeddings, for our dataset. We also noticed how employing **Xavier initialization** contributed to increasing the performance of the LSTM and BILSTM models.
The fine-tuned Arabert model is made publicly Available on the Huggingface Hub [1]

[1] huggingface.co/AraBert-finetuned-text-classification

4.3 Models explainability

4.3.1 BERT explainability

We utilized model-agnostic methods such as LIME, SHAP, and Integrated Gradients. For the transformer model, we focused on explaining the text tokens, as BERT relies on tokenization. In the case of USE and FastText, we explained the words since tokenization for these models primarily involves word splitting.

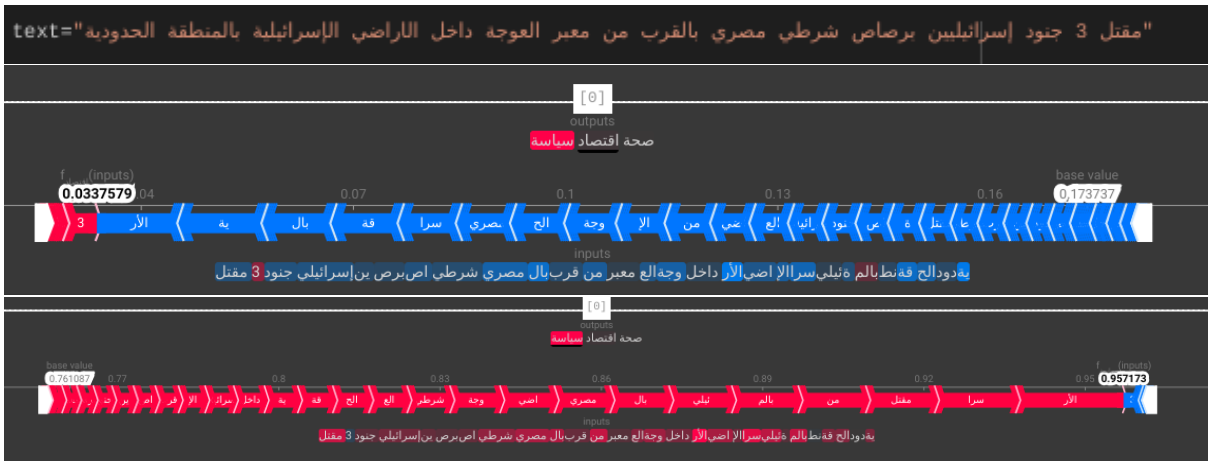


Figure 4.1: Shap Bert tokens explainability for BERT model, Notice how attached probabilities are important for the text predicted class سياسة

The SHAP plot demonstrates the token-level explainability for the BERT model. Notice how the attributed probabilities are significant for some of the text portion indicating the model's focus on these tokens for the first predicted class. This highlights the model's sensitivity to specific tokens within the input text.

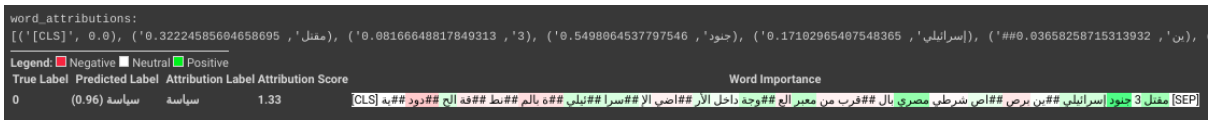


Figure 4.2: Integrated gradient explainability for BERT

The Integrated Gradients visualization further emphasizes the importance of individual tokens by displaying their contribution scores. This method highlights how each token's presence influences the model's prediction, providing a deeper understanding of the token interactions within the BERT model.

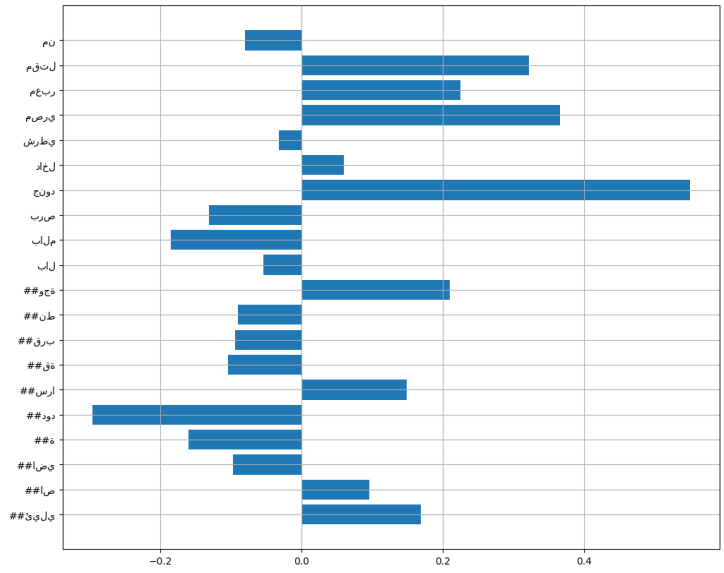


Figure 4.3: Integrated gradient explainability for BERT, tokens attributions visualized

The bar chart shows the Integrated Gradients explainability for BERT, where token attributions are visualized. Each bar represents the contribution of a token to the final prediction, offering a clear depiction of which tokens are most influential. This visual representation aids in interpreting the decision-making process of the BERT model and ensures that the model's predictions are transparent and interpretable.

4.3.2 LSTM explainability

We trained LSTM and BiLSTM models using both Universal Sentence Encoder (USE) and FastText embeddings. For FastText, we explained the contribution of each word. Since the TensorFlow API does not provide a method to access USE tokens, we explained the embeddings tensors instead of the tokens.

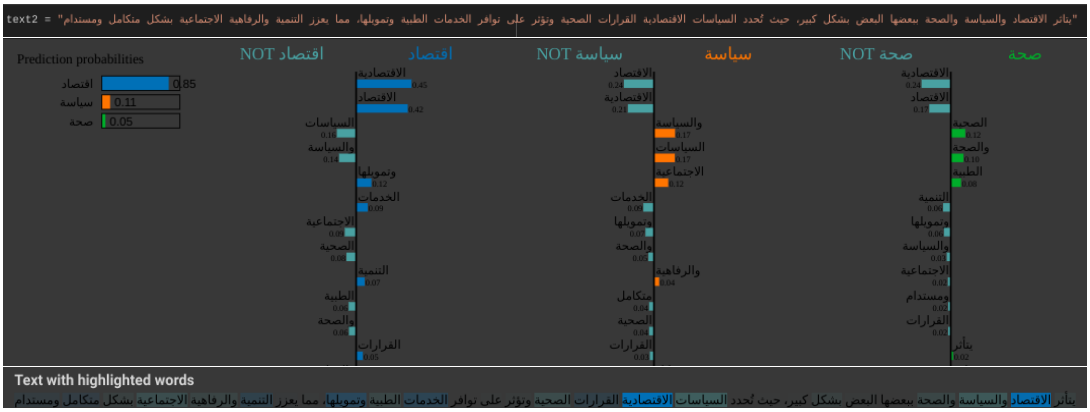
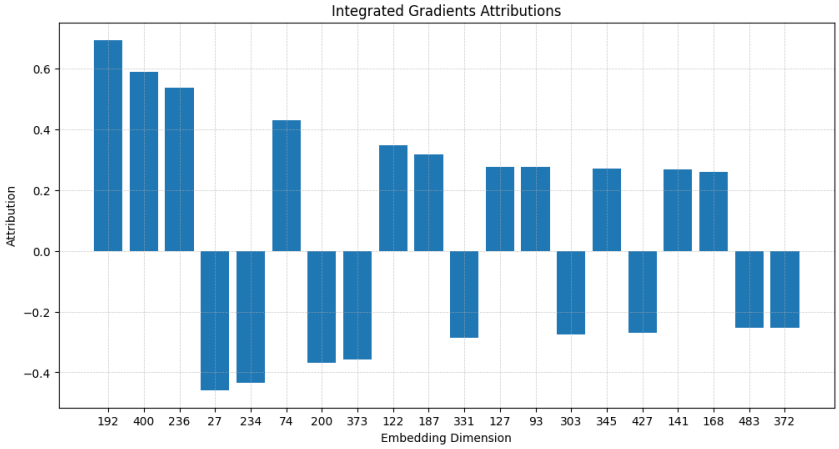


Figure 4.4: The lime explainer fits a regression line for every class, it displays the probabilties for a class against all other classes

It's worth mentioning that The LIME explainer fits a separate regression model for

each class to understand the local behavior of the original model. It displays the probabilities of the specified class in comparison to all other classes(complex model predictions probabilities for every class), thereby highlighting the contribution of each feature to the prediction for that class. That is why we visualize every single class attribution individually.




```
Top 20 important dimensions and their attributions:
Dimension 192: Attribution value 0.6950203635825123
Dimension 400: Attribution value 0.5900328293211773
Dimension 236: Attribution value 0.5386945006286631
Dimension 27: Attribution value -0.4585802596809476
Dimension 234: Attribution value -0.4334423939447214
Dimension 74: Attribution value 0.4312231134370122
Dimension 200: Attribution value -0.36891971225175385
Dimension 373: Attribution value -0.3566325905932296
Dimension 122: Attribution value 0.3483803475693096
Dimension 187: Attribution value 0.3173687567887845
Dimension 331: Attribution value -0.28539826312942473
Dimension 127: Attribution value 0.27599295622417225
Dimension 93: Attribution value 0.2759127891406197
Dimension 303: Attribution value -0.2752244840886422
Dimension 345: Attribution value 0.2719423234373439
Dimension 427: Attribution value -0.26986772341307314
Dimension 141: Attribution value 0.2695143102923651
Dimension 168: Attribution value 0.26087282941638346
Dimension 483: Attribution value -0.2541017905338588
Dimension 372: Attribution value -0.2532783257850616
Predicted label: ضحة
```

Figure 4.5: Integrated gradient on USE embeddings explainability, we have sorted the dimensions by their importance.

The input text explained is embedded to a tensor of shape (1, 512) and we have employed Integrated gradient to display the contribution of every value in the embedding tensor.

4.3.3 KNN explainability

We trained a K-Nearest Neighbors (KNN) model to classify Arabic text into three categories: Economy (اقتصاد), Politics (سياسة), and Health (صحة). For the purpose of interpretability, we employed the Local Interpretable Model-agnostic Explanations (LIME) method to explain the contribution of each word to the model's predictions.

LIME Explanation

To understand the KNN model's predictions, we utilized LIME to highlight the words in the text that contribute most significantly to the prediction. LIME works by fitting a separate regression model for each class to understand the local behavior of the original model.

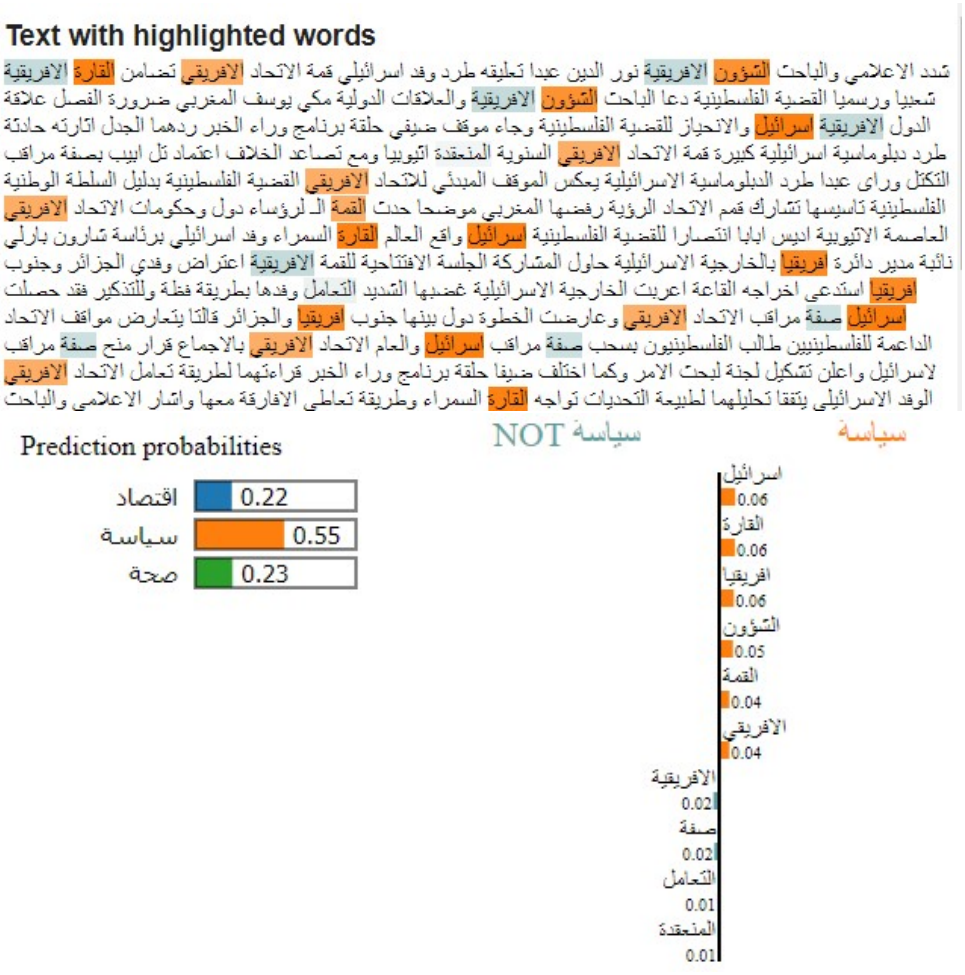


Figure 4.6: LIME Explanation of KNN Model Predictions

Key Contributing Words

The following words have the highest influence on the "Politics" prediction:

- إسرائيل (Israel)
- الإفريقية (African)
- الفلسطينية (Palestinian)
- الاتحاد (Union)
- النزاع (Conflict)

These words are highlighted in orange in the text, indicating their significant impact on the model's classification decision.

4.4 Discussion

Discussing the results of our experiments and their implications. The performance of various models using different embeddings methods is evaluated and compared. It is evident from Table 4.1 that the FastText embeddings significantly outperform USE embeddings across all model architectures. Specifically, the LSTM and BiLSTM models with FastText embeddings achieve the highest validation accuracy, indicating their effectiveness in capturing the nuances of Arabic text.

Moreover, the Arabert fine-tuned model also demonstrates high accuracy, showcasing the benefits of using pre-trained language models tailored to specific languages. The discrepancies in performance highlight the importance of selecting appropriate embeddings and model architectures based on the dataset and task at hand.

The application of hyper-parameter tuning using the Optuna package and the implementation of Xavier initialization for LSTM and BiLSTM models further enhance performance, suggesting that meticulous optimization can lead to significant improvements.

4.5 The Impact of XAI

Explainable AI (XAI) plays a crucial role in understanding and trusting machine learning models. By providing insights into the decision-making process of complex models,

XAI techniques such as LIME, SHAP, and Integrated Gradients help in identifying the contribution of individual features to the predictions.

In our study, the use of LIME for KNN, Integrated Gradients, and SHAP for BERT, LIME and integrated gradient for LSTM models allowed us to interpret the predictions effectively. These methods not only improve transparency but also help in diagnosing and mitigating biases in the models. For instance, the LIME explanations for the KNN model highlighted key words that significantly influenced the classification, which is essential for validating the model's behavior and ensuring fairness.

4.6 Challenges in Arabic Natural Language Processing

Arabic NLP faces several unique challenges due to the language's rich morphology, complex syntax, and diverse dialects. The following are some of the key challenges encountered:

- **Morphological Complexity**: Arabic words often have multiple forms due to prefixes, suffixes, and infixes, making tokenization and lemmatization difficult.
- **Dialectal Variations**: The presence of numerous dialects with significant differences poses a challenge for creating models that generalize well across all variants of Arabic.
- **Limited Resources**: There is a scarcity of annotated datasets and pre-trained models for Arabic compared to languages like English, which hampers the development of robust NLP applications.
- **Script and Orthographic Variations**: Arabic script can be written with or without diacritics, leading to ambiguity in text processing tasks.

Addressing these challenges requires a combination of linguistic expertise, data augmentation techniques, and the development of specialized tools and resources for Arabic NLP.

4.7 Conclusion and Perspectives

In conclusion, our study demonstrates the effectiveness of various models and embeddings methods for Arabic text classification. The significant performance gains achieved through the use of FastText embeddings and hyper-parameter tuning underscore the im-

portance of selecting appropriate techniques tailored to the specificities of the Arabic language.

Looking forward, there are several promising avenues for further research and development:

- **Enhanced Pre-trained Models**: Developing more comprehensive pre-trained models like Arabert for various Arabic dialects could improve performance and generalization.
- **Data Augmentation**: Leveraging data augmentation techniques to create larger and more diverse datasets can address the issue of limited resources.
- **Cross-lingual Transfer Learning**: Exploring transfer learning approaches from high-resource languages to Arabic can help mitigate the scarcity of annotated data.
- **Integrating XAI in Development**: Continuously integrating explainability methods throughout the model development process can ensure transparency and fairness, leading to more trustworthy AI systems.

Addressing these challenges and exploring new research directions, would help significantly advance the field of Arabic NLP and create more effective and equitable language technologies.

Bibliography

[1] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier.

[2] Scott M. Lundberg A , Su-In Lee (2017) Unified Approach to Interpreting Model Predictions

[3] Mukund Sundararajan, Ankur Taly, Qiqi Yan (2017) Axiomatic Attribution for Deep Networks

[4] Universal Sentence Encoder Daniel Cera, Yinfei Yanga, Sheng-yi Konga, Nan Huaa , Nicole Limtiacob , Rhomni St. Johna, Noah Constanta , Mario Guajardo-Cespedes´a , Steve Yuanc , Chris Tara , Yun-Hsuan Sunga , Brian Stropea , Ray Kurzweila

[5] Enriching Word Vectors with Subword Information Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov

Appendix

Figure 7: An illustration of optuna trials for hyper-parameters-tuning

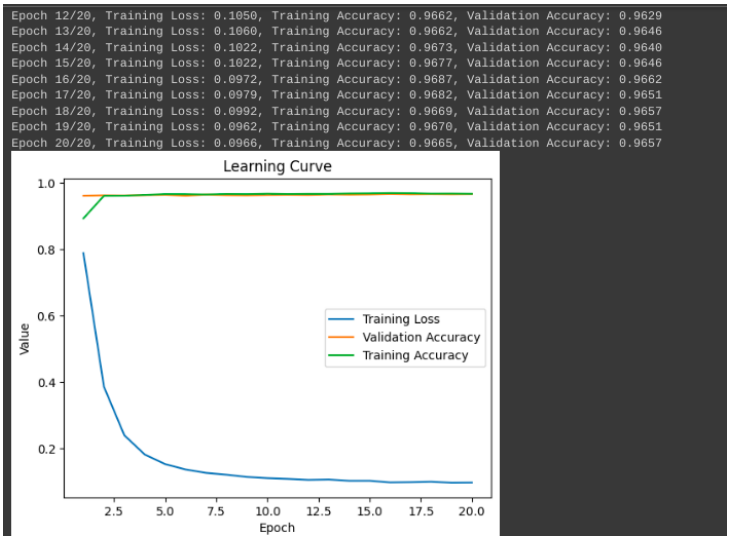


Figure 8: Learning curve for LSTM model with fasttext embeddings

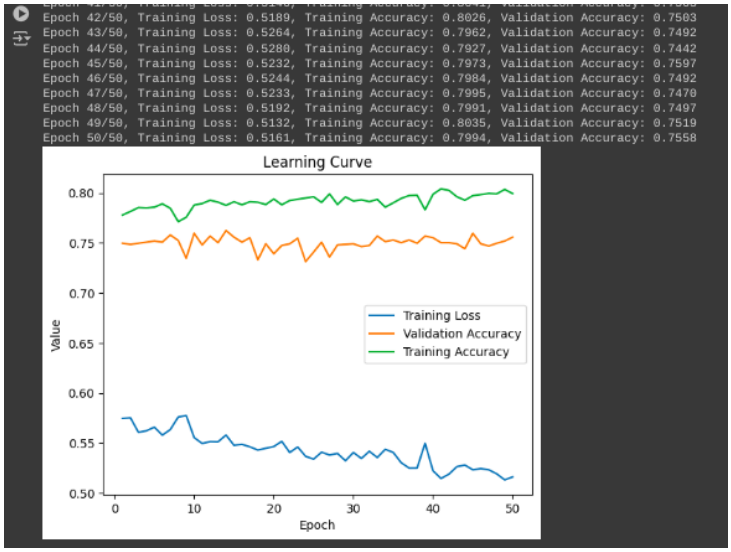


Figure 9: Learning curve for LSTM model with USE embeddings

Training Loss	Epoch	Step	Accuracy	Validation Loss	Macro F1	Recall
No log	0.99	56	0.9430	0.1530	0.9427	0.9425
No log	1.99	112	0.9579	0.1230	0.9577	0.9577
No log	3.0	169	0.9607	0.1287	0.9605	0.9608
No log	3.99	225	0.9618	0.1296	0.9616	0.9618
No log	5.0	282	0.9623	0.1147	0.9623	0.9622
No log	5.99	338	0.9612	0.1500	0.9611	0.9612
No log	7.0	395	0.9601	0.1953	0.9599	0.9602
No log	7.99	451	0.9640	0.1713	0.9639	0.9640
0.0526	9.0	508	0.9646	0.1748	0.9644	0.9645
0.0526	9.92	560	0.9646	0.1768	0.9644	0.9645

Figure 10: Learning curve for fine-tuning AraBERT Model

Résumé

Notre projet, intitulé **"Explainable AI for Arabic Web Pages Classification"**, applique des techniques de **traitement du langage naturel (NLP)** pour la classification de text Arabe. Nous avons exploré des modèles qui sont des variants des **réseaux de neurones artificiels (ANN)**, y compris les réseaux **Long Short-Term Memory (LSTM)** et les **Bidirectional Encoder Representations from Transformers (BERT)**. En utilisant diverses bibliothèques et frameworks, nous avons comparé les performances de chaque modèle afin d'identifier le plus efficace.

Nos expériences ont montré les précisions de validation suivantes : **Arabert** finetuned a atteint 96.23%, **LSTM** et **BILSTM** avec l'embedding **USE** ont atteint 75.23%, **KNN** avec l'embedding USE a atteint 69.00%, **LSTM-BILSTM** avec l'embedding **Fasttext** a atteint 96.75%, et KNN avec l'embedding Fasttext a atteint 95.70%.

Nous avons également intégré des techniques d'**Intelligence Artificielle Explicable (XAI)**, y compris **LIME**, **SHAP** et **Integrated Gradients**, pour améliorer la transparence et l'interprétabilité de nos modèles. Ces méthodes XAI nous ont permis de fournir des explications détaillées pour les prédictions de notre modèle, favorisant ainsi la confiance et permettant une prise de décision plus éclairée. Grâce à nos efforts, nous avons cherché à contribuer au domaine croissant de l'IA explicable, en particulier pour la classification des pages web arabes.