PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

Ministry of Higher Education and Scientific Research

University of Mohamed El-Bachir El-Ibrahimi – Bordj Bou Arreridj

Faculty of Mathematics and Computer Science



MASTER'S THESIS

Submitted in partial fulfillment of the requirements for the degree of

Master in Computer Science

Specialization: Information and Communication Technologies (ICT)

TITLE

Sentiment Analysis of Algerian Darija Comments on Social Media

---

Presented by:
BAKHA Yamina
BENDJEDDOU Soumaya Amira

Defense date: 30 June 2026

The jury composed of:

Dr. NOUIOUA Farid	President
Dr. BOUZIANE Abderaouf	Examiner
Dr. BOUNAGUEB Adel	Rep./Incubateur
Dr. BEGHOURA Mohamed Amine	Supervisor

Academic Year 2025–2026

Remerciements

Alhamdulillah, all praises to ALLAH for His infinite greatness, blessings, and divine grace, for granting us the patience, courage, and strength to endure the challenges of this academic journey and successfully complete this thesis work.

First and foremost, it is with immense pleasure and profound gratitude that we express our sincerest thanks and highest appreciation to our supervisor, **Mr. BEGHOURA Mohamed Amine**. We are deeply indebted to him for his unwavering support, constant availability, and invaluable guidance throughout every stage of this research. His constructive feedback, rigorous academic standards, and the wealth of knowledge he shared with us have been fundamental to the realization of this work.

We would like to express our special gratitude and heartfelt appreciation to **Fatima**, **Mohammed**, and **Tayeb**. Their immense help, technical assistance, valuable time, and constant encouragement were crucial in overcoming many difficulties during this journey, and we are truly grateful for their support.

Finally, we extend our deepest respect and gratitude to the distinguished members of our evaluation jury. We thank **Mr. NOUIOUA FARID** (President), **Mr. BOUZIANE ABDERAOUF** (Examiner), and **Mr. BOUNAGUEB ADEL** (Rep./Incubateur) for their time, interest, and for kindly agreeing to review and evaluate our work. Their insightful recommendations and expertise are highly valued and will undoubtedly help refine the quality of this research.

Dedication

With profound gratitude, I first and foremost thank Allah, the Almighty, for granting me the strength, patience, and guidance to complete this work.

I dedicate this achievement to my beloved **Mother** and **Father**, whose unconditional love, endless sacrifices, heartfelt prayers, and unwavering support have been the foundation of every success in my life. Thank you for always believing in me, encouraging me to pursue my dreams, and standing beside me through every challenge. This accomplishment is as much yours as it is mine.

To my beloved sister **Fatima**, my soulmate and closest friend, thank you for your unconditional love, constant encouragement, and unwavering support. I am truly grateful to have you by my side, and I am incredibly proud of the remarkable woman you have become and everything you continue to achieve.

To my beloved sister **Latifa**, my dear **Auntie Meriem**, and my beloved **brothers**, thank you for your love, prayers, encouragement, and continuous support. Your kindness and belief in me have always motivated me to keep moving forward, and I am truly blessed to have each of you in my life.

To my dear cousin, my best friend, and my beloved fiancé, **Tayeb**, thank you for your endless love, patience, encouragement, and unwavering belief in me. Your comforting words, constant support, and presence gave me the strength to overcome every challenge. I will always be grateful for everything you have done for me.

To my beloved **grandmother** and to the cherished memory of my late **grandfather**, thank you for your love, prayers, wisdom, and inspiration. Although you are no longer with us, your memory continues to guide me every day.

Finally, I would like to express my sincere gratitude to my teachers, **Ben Tayeb Madani**, **Ben Nacef Abdelhak**, **Dr.MOUSSAOUI Boubakeur**,**Dr.BENAOUDA Nadjib** and **Dr.CHARIKHI Mourad**, as well as to all the teachers who have guided, inspired, and supported me throughout my academic journey. Thank you for your dedication and for sharing your knowledge.

Soumaya Amira

Dedication

(وَقُلِ اعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ وَالْمُؤْمِنُونَ)
صدق الله العظيم

(رَبِّ أَوْزِعْنِي أَنْ أَشْكُرَ نِعْمَتَكَ الَّتِي أَنْعَمْتَ عَلَيَّ وَعَلَىٰ وَالِدَيَّ وَأَنْ أَعْمَلَ صَالِحًا تَرْضَاهُ)
صدق الله العظيم

الحمد لله الذي بنعمته تتم الصالحات، والذي وفقني وأعانني على إتمام هذا العمل، والصلاة والسلام على سيدنا محمد صلى الله عليه وسلم.
أهدي ثمرة هذا الجهد المتواضع...

إلى أبي الغالي...

إلى من كان قدوتي في الصبر والكفاح، إلى الرجل الذي علمني أن النجاح لا يأتي إلا بالإصرار والعمل، والذي بذل الغالي والنفيس من أجل أن أصل إلى هذه اللحظة. كلمات الشكر تعجز عن وصف مقدار امتناني لك، فلك مني كل الحب والتقدير، وأسأل الله أن يحفظك، ويبارك في عمرك، ويجزيك عني خير الجزاء.

إلى أمي الحبيبة...

إلى من جعل الله الجنة تحت قدميها، إلى القلب الذي احتواني، والعين التي سهرت من أجلي، والدعاء الذي كان يرافقني في كل خطوة. مهما كتبت فلن أوفيك حقك، فأنتِ سر نجاحي بعد الله، وأجمل نعمة أنعم الله بها عليّ. أسأل الله أن يديمك نورًا في حياتي، وأن يرزقك الصحة والسعادة وطول العمر.

إلى إخوتي الأعزاء...

إلى من كانوا خير سندٍ وعونٍ لي في مسيرتي، وإلى من تقاسمت معهم أجمل الذكريات وأصعب اللحظات. شكرًا لكم على تشجيعكم الدائم، ووقوفكم إلى جانبي، ودعمكم الذي منحني القوة لمواصلة الطريق. أسأل الله أن يحفظكم، ويوفقكم في حياتكم، ويجمعنا دائمًا على المحبة والخير.

إلى أختي الغالية وأبنائها الأحبة...

إلى رفيقة العمر، وشريكة الذكريات الجميلة، وسندي في أوقات الفرح والتعب إلى أختي التي كانت دائمًا قريبة من قلبي، وسندي في أوقات الفرح والتعب. شكرًا لكل كلمة تشجيع، ولكل دعاء صادق، ولكل لحظة دعم منحتنني فيها القوة للاستمرار وإلى أطفالها الذين كانوا يرسمون البهجة في أيامي ويخففون عني عناء الدراسة. أسأل الله أن يحفظكم جميعًا، وأن يوفقكم، ويجعل أيامكم مليئة بالخير والنجاح.

إلى زوجة أخي العزيزة...

شكرًا لكِ على طيب قلبك، وصدق مشاعرك، وحسن معاملتك، وعلى كل كلمات التشجيع والدعم التي قدمتها لي. أسأل الله أن يديم عليكم المحبة والمودة، ويرزقكم حياة مليئة بالخير والبركة.

إلى كل من أحبني بصدق...

إلى كل من وقف إلى جانبي، ودعا لي، وآمن بقدرتي على النجاح، وإلى كل من كان له أثر طيب في رحلتي العلمية والإنسانية، أهديكم هذا الإنجاز المتواضع، راجيًا من الله أن يجعله بدايةً لمستقبل مليء بالنجاح والعطاء.

(وَمَا تَوْفِيقِي إِلَّا بِاللَّهِ عَلَيْهِ تَوَكَّلْتُ وَإِلَيْهِ أُنِيبُ)

أمينة

AI Usage Disclaimer

Artificial intelligence tools were used only to support language refinement, formatting, organization, and technical assistance during the preparation of this work. The scientific content, methodology, experiments, analysis, results, and final decisions remain the responsibility of the students.

Abstract

Sentiment analysis has become an important task in Natural Language Processing (NLP), especially for understanding public opinions expressed on social media. However, Algerian Arabic remains a low-resource dialect due to the lack of annotated corpora, linguistic tools, and standardized writing forms. This work focuses on sentiment analysis of Algerian Darija comments collected from social media platforms, mainly YouTube and Twitter. In this study, we built and prepared the *AlgerianPulse* corpus through several stages, including data collection, cleaning, preprocessing, annotation, dataset integration, and balancing. The comments were classified into three sentiment categories: positive, negative, and neutral. Different approaches were investigated, including traditional machine learning models, word embedding techniques such as Word2Vec and FastText, deep learning models, pretrained transformer-based language models, and a Hard Majority Voting ensemble. The experimental results show that pretrained transformer models outperformed the traditional machine learning and deep learning approaches, with DziriBERT achieving the best individual performance. Furthermore, the proposed Hard Majority Voting ensemble achieved the highest overall performance, reaching an accuracy of 86.42% and demonstrating the effectiveness of combining multiple classifiers to improve the robustness and reliability of sentiment prediction. Overall, this work contributes to the development of sentiment analysis resources and models for Algerian Arabic. It highlights the challenges of dialectal Arabic processing, particularly spelling variation, code-switching, informal expressions, and the scarcity of labeled data, while proposing a practical methodology for building and evaluating sentiment classification systems in a low-resource language context.

Keywords: Sentiment Analysis, Algerian Arabic, Darija, Natural Language Processing, Low-Resource Languages, Machine Learning, Deep Learning, Pretrained Transformers, Hard Majority Voting, Social Media.

Résumé

L'analyse des sentiments est devenue une tâche importante dans le traitement automatique du langage naturel (TALN), en particulier pour comprendre les opinions publiques exprimées sur les réseaux sociaux. Cependant, l'arabe algérien reste un dialecte à faibles ressources en raison du manque de corpus annotés, d'outils linguistiques et de formes d'écriture standardisées. Ce travail se concentre sur l'analyse des sentiments des commentaires en darija algérienne collectés à partir de plateformes de médias sociaux, principalement YouTube et Twitter. Dans cette étude, nous avons construit et préparé le corpus *AlgerianPulse* à travers plusieurs étapes, notamment la collecte de données, le nettoyage, le prétraitement, l'annotation, l'intégration d'ensembles de données et l'équilibrage. Les commentaires ont été classés en trois catégories de sentiments : positif, négatif et neutre. Différentes approches ont été explorées, y compris les modèles d'apprentissage automatique traditionnels, les techniques de plongement de mots (word embedding) telles que Word2Vec et FastText, les modèles d'apprentissage profond, les modèles de langage pré-entraînés basés sur les transformeurs, et une stratégie finale de vote majoritaire. Les résultats expérimentaux montrent que les modèles de transformeurs pré-entraînés ont surpassé les approches traditionnelles d'apprentissage automatique et d'apprentissage profond. Parmi les modèles individuels, **DziriBERT** a obtenu les meilleures performances. En outre, l'ensemble proposé basé sur le *Hard Majority Voting* a atteint les meilleures performances globales avec une précision (*Accuracy*) de **86,42 %**, démontrant l'efficacité de la combinaison de plusieurs classifieurs pour améliorer la robustesse et la fiabilité de la prédiction des sentiments. Globalement, ce travail contribue au développement de ressources et de modèles d'analyse des sentiments pour l'arabe algérien. Il met en évidence les défis du traitement de l'arabe dialectal, en particulier la variation orthographique, l'alternance codique (code-switching), les expressions familières et la rareté des données étiquetées, tout en proposant une méthodologie pratique pour la construction et l'évaluation de systèmes de classification des sentiments dans un contexte de langue à faibles ressources.

Mots-clés : Analyse des sentiments, Arabe algérien, Darija, Traitement automatique du langage naturel, Langues à faibles ressources, Apprentissage automatique, Apprentissage profond, Transformeurs pré-entraînés, Hard Majority Voting, Réseaux sociaux.

ملخص

أصبح تحليل المشاعر مهمة أساسية في معالجة اللغات الطبيعية، لا سيما لفهم الآراء العامة المعبر عنها عبر وسائل التواصل الاجتماعي. ومع ذلك، لا تزال العربية الجزائرية لهجة ذات موارد محدودة نظراً لافتقارها إلى المدونات النصية المشروحة، الأدوات اللغوية، والصيغ الكتابية الموحدة. يركز هذا العمل على تحليل المشاعر للتعليقات بالدارجة الجزائرية التي تم جمعها من منصات التواصل الاجتماعي، وبشكل أساسي يوتيوب وتويتر.

في هذه الدراسة، قمنا ببناء وإعداد مدونة AlgerianPulse النصية عبر عدة مراحل، شملت جمع البيانات، التنظيف، المعالجة الأولية، الوسم، دمج البيانات، وموازنتها. تم تصنيف التعليقات إلى ثلاث فئات للمشاعر: إيجابي، سلبي، ومحايد. كما تم استكشاف مقاربات مختلفة، بما في ذلك نماذج التعلم الآلي التقليدية، تقنيات تمثيل الكلمات word embedding مثل Word2Vec و FastText، نماذج التعلم العميق، نماذج اللغة سابقة التدريب المعتمدة على المحولات (Transformers)، بالإضافة إلى استراتيجية التصويت بالأغلبية في المرحلة النهائية.

أظهرت النتائج التجريبية أن نماذج المحولات سابقة التدريب تفوقت على أساليب التعلم الآلي التقليدية والتعلم العميق، حيث حقق نموذج DziriBERT أفضل أداء بين النماذج الفردية. إضافةً إلى ذلك، حقق أسلوب التجميع المقترح القائم على التصويت بالأغلبية أفضل أداء إجمالي، إذ بلغ معدل الدقة **86.42%**، مما يؤكد فعالية دمج عدة نماذج لتحسين متانة وموثوقية التنبؤ النهائي بالمشاعر.

بشكل عام، يساهم هذا العمل في تطوير موارد ونماذج تحليل المشاعر للغة العربية الجزائرية. كما يسلط الضوء على تحديات معالجة العربية اللهجية، وخاصة التباين الإملائي، التناوب اللغوي (code-switching)، التعبيرات غير الرسمية، وندرة البيانات المصنفة، مع اقتراح منهجية عمل علمية لبناء وتقييم أنظمة تصنيف المشاعر في سياق اللغات ذات الموارد المحدودة.

الكلمات المفتاحية: تحليل المشاعر، العربية الجزائرية، الدارجة، معالجة اللغات الطبيعية، اللغات ذات الموارد المحدودة، التعلم الآلي، التعلم العميق، المحولات سابقة التدريب، التصويت بالأغلبية، وسائل التواصل الاجتماعي.

Liste des figures

2.1 Illustration of the Support Vector Machine (SVM) decision boundaries separating different classes. . . . 17
2.2 Architecture of the Bidirectional Long Short-Term Memory (BiLSTM) network. . . . 18
2.3 Architecture of the Transformer encoder. . . . 19

4.1 Workflow diagram of our proposed approach. . . . 32
4.2 Corpus Construction Framework (generated with AI assistance). . . . 33
4.3 Automatic sentiment annotation process using Label Studio and locally deployed large language models. . . . 34
4.4 Streamlit application for manual review and correction of test-set annotations. . . . 35
4.5 Illustration of the Hard Voting Ensemble Method. . . . 42

5.1 Final Sentiment Distribution 47
5.2 Top 10 Positive Bigrams 48
5.3 Top 10 Negative Bigrams 48
5.4 Top 10 Positive Trigrams 49
5.5 Top 10 Negative Trigrams 50
5.6 Positive Word Cloud 51
5.7 Negative Word Cloud 51
5.8 Bar chart accuracy comparison of machine learning models 52
5.9 Bar chart accuracy comparison of deep learning models 53
5.10 Bar chart accuracy comparison of pretrained transformer models 55
5.11 Confusion Matrix of the DziriBERT Model on the Test Set 56
5.12 Confusion Matrix of the Hard Majority Voting Ensemble 59

6.1 AlgerianPulse Three-Tier System Architecture (generated with AI assistance) 63
6.2 User use case diagram of the AlgerianPulse platform 65
6.3 Administrator use case diagram of the AlgerianPulse platform 66
6.4 Class diagram of the AlgerianPulse platform 66
6.5 Homepage of the AlgerianPulse platform 67
6.6 User authentication interface of the AlgerianPulse platform 68
6.7 User dashboard of the AlgerianPulse platform 69
6.8 Social Account Management Interface 70

6.9 Posts Management Interface 71
6.10 CSV Dataset Upload Interface 72
6.11 Sentiment Analysis Interface 72
6.12 Report Generation Interface 73
6.13 Administrator Dashboard 74

Liste des tableaux

3.1 Example of Aspect-Level Sentiment Analysis 26
3.2 Comparative Table of Related Work 29

4.1 Parameters of the TF-IDF Feature Extraction Methods. 36

5.1 Main Statistics of the AlgeriaPulse Unlabeled Corpus 46
5.2 Evolution of the Labeled Corpus 46
5.3 Class Distribution of the Training, Validation, and Test Sets 47
5.4 Main Statistics of the Generated Text Representations 52
5.5 Class-wise Performance of DziriBERT 55
5.6 Representative Misclassified Comments by DziriBERT 57
5.7 Class-wise Performance of the Hard Majority Voting Ensemble 58
5.8 Overall Performance Comparison of All Sentiment Classification Models 60

Liste des abréviations

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
BoW	Bag-of-Words
CBOW	Continuous Bag-of-Words
CNN	Convolutional Neural Network
DL	Deep Learning
ELMo	Embeddings from Language Models
IDF	Inverse Document Frequency
LR	Logistic Regression
LRL	Low-Resource Language
LRLs	Low-Resource Languages
ML	Machine Learning
MSA	Modern Standard Arabic
NLP	Natural Language Processing
SA	Sentiment Analysis
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency–Inverse Document Frequency

Table des matières

Remerciements **I**

Dedication **II**

Dedication **III**

AI Usage Disclaimer **IV**

Abstract **V**

Résumé **VI**

ملخص **VII**

Liste des figures **IX**

Liste des tableaux **X**

Liste des abréviations **XI**

General Introduction **1**

1 Context and Motivation 1
2 Problem Statement 1
3 Objectives and Contributions 2
4 Dissertation Organisation 2

1 Low-Resources Languages and the Case of the Algerian Dialect **4**

1.1 Low-Resource Languages 4
1.2 Examples of Low-Resource Languages 4
1.3 Challenges of Low-Resource Languages 5
1.3.1 Data Scarcity 5
1.3.2 Lack of Linguistic Resources and NLP Tools 5
1.3.3 Machine Translation Challenges 6
1.3.4 Linguistic Complexity 6
1.3.5 Orthographic Variation and Code-Switching 6
1.4 The Algerian Dialect 6
1.4.1 Characteristics of the Algerian Dialect 7
1.5 Why is Algerian Dialect a Low-Resource Language 9

1.6 Conclusion 9

2 Text Representation and Classification Methods 11
2.1 Natural Language Processing 11
2.2 traditional-text-representation-methods 12
2.2.1 TF-IDF 12
2.2.2 Character TF-IDF 12
2.2.3 Hybrid TF-IDF of Words and Characters 13
2.3 Word Embedding Methods 13
2.3.1 Word2Vec 13
2.3.2 FastText: 14
2.4 document-representation-methods 15
2.5 Classification Models 16
2.5.1 Machine Learning 16
2.5.2 Support Vector Machine (SVM) 16
2.5.3 Logistic Regression 17
2.6 Deep learning 17
2.6.1 Bidirectional Long-Term Memory (BiLSTM) 18
2.6.2 Transformer Encoder 18
2.7 Pretrained Transformer Models 21
2.7.1 AraBERT 21
2.7.2 MARBERT 21
2.7.3 DziriBERT 21
2.7.4 XLM-RoBERTa 22

3 Sentiment Analysis 23
3.1 Fundamentals of Sentiment Analysis 23
3.1.1 Definition of Sentiment Analysis 23
3.1.2 Opinion, Emotion, and Subjectivity 23
3.1.3 Objective and Subjective Texts 24
3.1.4 Explicit and Implicit Opinions 25
3.2 Levels of Sentiment Analysis 25
3.2.1 Document-Level SA 25
3.2.2 Sentence-Level SA 25
3.2.3 Aspect-Level SA 25
3.3 Sentiment Classification Categories 26
3.3.1 Binary Classification 26
3.3.2 Multi-Class Classification 26
3.3.3 Fine-Grained Classification 26
3.4 Challenges of Sentiment Analysis in the Algerian Dialect 27

3.4.1 Dialectal Variations . . . 27
3.4.2 Code-Switching . . . 27
3.4.3 Arabizi Writing . . . 27
3.4.4 Orthographic Variations . . . 28
3.4.5 Resource Scarcity . . . 28
3.4.6 Sarcasm and Figurative Language . . . 28
3.5 Related Work . . . 29

4 Methodology **31**
4.1 Methodological Workflow . . . 31
4.2 AlgeriaPulse Dataset Construction . . . 32
4.2.1 Data Sources . . . 33
4.2.2 Corpus Construction . . . 33
4.2.3 Annotation . . . 34
4.2.4 Data Preprocessing . . . 34
4.3 Feature Extraction and Word Embedding Techniques . . . 36
4.3.1 TF-IDF-Based Representations . . . 36
4.3.2 Word2Vec Embeddings . . . 37
4.3.3 FastText Embeddings . . . 37
4.3.4 Doc2Vec Embeddings . . . 37
4.3.5 Hybrid Representation . . . 37
4.4 Model Training . . . 38
4.4.1 Machine Learning Models . . . 38
4.4.2 Deep Learning Models . . . 38
4.4.3 Transformer-Based Models . . . 39
4.5 Evaluation Metrics . . . 40
4.6 Majority Voting . . . 41
4.6.1 Hard Voting . . . 41
4.6.2 Soft Voting . . . 41
4.6.3 Voting Strategy Adopted in This Work . . . 41

5 Implementation and Results **44**
5.1 Experimental Environment . . . 44
5.1.1 Hardware Resources . . . 44
5.1.2 Programming Language . . . 44
5.1.3 Libraries and Frameworks . . . 44
5.2 Corpus Analysis . . . 46
5.2.1 Corpus Statistics . . . 46
5.2.2 Bigram Analysis . . . 48
5.2.3 Trigram Analysis . . . 49

5.2.4 Word Cloud Visualization 50
5.3 Representation Statistics 51
5.4 Experimental Results 52
5.4.1 Machine Learning Models 52
5.4.2 Deep Learning Models Performance 53
5.4.3 Pretrained Transformer Models Performance 54
5.4.4 Performance Analysis of the Best Individual Model 55
5.5 Error Analysis 56
5.6 Performance Evaluation of the Hard Majority Voting Ensemble 58
5.6.1 Overall Model Comparison 59
5.7 Conclusion 61

6 Architecture and Design of the AlgerianPulse Platform 62
6.1 System Architecture 62
6.2 Development Environment 63
6.2.1 Frontend Technologies 63
6.2.2 Backend Technologies 64
6.2.3 AI Technologies 64
6.2.4 Development Tools 64
6.3 UML Design 65
6.3.1 Use Case Diagram 65
6.3.2 Class Diagram 65
6.4 User Interface Implementation 66
6.4.1 Homepage 67
6.4.2 User Authentication 67
6.4.3 User Dashboard 68
6.4.4 Social Account Management 69
6.4.5 Posts Management 70
6.4.6 CSV Dataset Upload 71
6.4.7 Post Analysis 72
6.4.8 Report Generation 73
6.5 Administrator Dashboard 73
6.6 Integration of the Sentiment Analysis Models 74
6.6.1 AI Model Deployment 74
6.6.2 Gemini API Integration 75
6.6.3 Hard Majority Voting Integration 75
6.6.4 Prediction Storage and Visualization 75
6.7 Conclusion 75

General Conclusion 76

References **78**

General Introduction

1 Context and Motivation

In recent years, social media platforms have become important spaces where people express their opinions, emotions, and reactions toward different topics, products, services, events, and public issues. These large amounts of user-generated content represent a valuable source of information for understanding public sentiment. For this reason, sentiment analysis has become one of the most important tasks in Natural Language Processing (NLP), as it aims to automatically identify whether a text expresses a positive, negative, or neutral opinion.

However, most sentiment analysis systems are developed for high-resource languages such as English, French, or Modern Standard Arabic. In contrast, Algerian Arabic, commonly known as Algerian Darija, remains a low-resource dialect. It is widely used in daily communication and on social media, but it does not have a standardized writing system. Users often mix Arabic, French, English, Arabizi, emojis, slang expressions, and different spelling forms in the same comment. This makes automatic sentiment classification more difficult.

The motivation behind this work is to contribute to the development of sentiment analysis resources and models for Algerian dialectal Arabic. By building and evaluating different classification approaches, this study aims to support the processing of Algerian social media comments and highlight the challenges related to low-resource dialects.

2 Problem Statement

Although Algerian Darija is widely used online, it remains underrepresented in NLP research. The main problem is the lack of large annotated datasets, linguistic resources, and pretrained models specifically adapted to Algerian dialect. In addition, Algerian social media comments are highly informal and noisy. They may contain spelling variations, code-switching, abbreviations, emojis, repeated letters, and non-standard grammatical structures.

These characteristics make sentiment analysis more challenging, especially when using models trained on standard Arabic or other dialects. Therefore, there is a need to build a suitable corpus and evaluate different machine learning, deep learning, and transformer-based models for Algerian dialect sentiment classification.

3 Objectives and Contributions

The main objective of this work is to develop and evaluate sentiment analysis models for Algerian Darija comments collected from social media platforms. The comments are classified into three sentiment categories: positive, negative, and neutral.

The main contributions of this dissertation are:

- Building and preparing the AlgeriaPulse corpus for Algerian dialect sentiment analysis.
- Cleaning, preprocessing, and balancing social media comments from different sources.
- Applying traditional machine learning methods for sentiment classification.
- Training and evaluating word embedding-based approaches such as Word2Vec and FastText.
- Evaluating deep learning models and pretrained transformer-based language models.
- Comparing the performance of different models using standard evaluation metrics.
- Proposing and evaluating a Hard Majority Voting ensemble to combine the predictions of several sentiment classification models.
- Highlighting the main challenges of Algerian dialect processing in a low-resource context.

4 Dissertation Organisation

This dissertation is organized into six chapters.

Chapter 1, entitled *Low-Resources Languages and the Case of the Algerian Dialect*, introduces the concept of low-resource languages and explains the specific case of Algerian Darija. It presents its linguistic characteristics, writing variations, and the main challenges related to its automatic processing.

Chapter 2, entitled *Text Representation and Classification Methods*, presents the main techniques used to represent and classify textual data. It discusses Natural Language Processing, traditional text representation methods, word embedding techniques, document representation methods, machine learning classifiers, deep learning models, and pretrained transformer models.

Chapter 3, entitled *Sentiment Analysis*, introduces the main concepts of sentiment analysis. It explains sentiment classification levels and categories, discusses the

challenges of sentiment analysis in Algerian dialect, and presents related work in this field.

Chapter 4, entitled *Methodology*, describes the methodological workflow followed in this work. It explains the construction of the AlgeriaPulse dataset, data collection, annotation, preprocessing, feature extraction, model training, ... evaluation metrics, and the proposed Hard Majority Voting ensemble.

Chapter 5, entitled *Implementation and Results*, presents the experimental environment and the obtained results. It includes corpus analysis, representation statistics, the evaluation and comparison of the different sentiment classification models, and the performance of the proposed Hard Majority Voting ensemble.

Chapter 6, entitled *Architecture and Design of the AlgerianPulse Platform*, presents the prototype platform developed in this project. It describes the system architecture, development environment, system design, user interface implementation, and integration of the sentiment analysis models.

Finally, the dissertation ends with a general conclusion that summarizes the main findings, discusses the limitations of the work, and suggests possible future improvements.

Chapter 1

Low-Resources Languages and the Case of the Algerian Dialect

Introduction

This chapter explores the concept of Low-Resource Languages (LRLs) in NLP, detailing their main challenges and providing real-world examples. Following this general overview, the focus shifts to the Algerian dialect to analyze its specific linguistic features, including lexical diversity, regional variations, and phonological characteristics. Lastly, the chapter investigates the factors that make the Algerian dialect a low-resource language. It highlights the major obstacles it creates for NLP applications, such as sentiment analysis and machine translation, which stem primarily from code-switching, unstandardized spelling, and a lack of formal standardization.

1.1 Low-Resource Languages

Low-resource languages are languages that lack sufficient computational and linguistic resources, including limited pretrained models, annotated corpora, and lexical databases. This scarcity of resources affects the performance of Natural Language Processing (NLP) tasks negatively, such as speech recognition, machine translation, sentiment analysis, and text classification[1]. There is no universal standard for defining a low-resource language. In general, a language is considered a limited resource when it lacks the annotated datasets, linguistic tools, digital resources, and computational models necessary to develop robust NLP applications.

Ultimately, the notion of resource availability is relative and depends on the quantity and quality of existing linguistic data, not on the number of language speakers. For instance, a language with millions of speakers but lacking sufficient natural language processing tools and datasets could be classified as low-resource language. Conversely, a language spoken by only a few people may not be considered low-resource if sufficient linguistic resources are developed.

1.2 Examples of Low-Resource Languages

Over 90% of the world's 7,164 distinct languages are classified as low-Resource Languages (LRLs) for Natural Language Processing (NLP) research[2]. Classifying a language as LRL is not necessarily determined by the number of its speakers, but rather by the availability of annotated datasets, pre-trained models, and computational tools.

For example, Hindi is spoken by 600 million people worldwide, yet it is considered low-resource. Icelandic, on the other hand, is spoken by approximately 360,000 people and is not considered an LRL due to the availability of high-quality annotated corpora, advanced NLP tools, and a strong research support.

Low-Resource Languages are found across all continents. Indigenous languages of the Americas, including Colombian indigenous languages, suffer from limited linguistic resources [3]. In Asia, languages such as Assamese, Bengali, and Bodo also have limited resources problems in the NLP applications[4]. Europe also includes Low-Resource Languages such as Irish and Basque (spoken in Spain and France), which still face limited resources compared to high-resource languages. Similarly, several African languages such as the Nigerian[5] and the Malian languages, suffer from a lack of linguistic resources. In addition to languages, numerous dialects worldwide suffer from the same challenges, such as a lack of annotated data, pre-trained models, data scarcity, and small digital datasets. An example of this is the Algerian dialect, which requires the manual creation and annotation of datasets for NLP tasks.

All these examples demonstrate that languages and dialects with limited resources are spread across diverse geographic regions and linguistic families. Despite their diversity, they share common challenges such as data scarcity, limited computational resources, and inadequate natural language processing tools.

1.3 Challenges of Low-Resource Languages

Natural language processing applications for limited-resource languages face several challenges that restrict their performance and adoption. These challenges arise from linguistic and technical factors, which can affect numerous tasks such as machine translation, sentiment analysis, text classification, and speech recognition.

1.3.1 Data Scarcity

Data scarcity is the most significant challenge faced by LRLs. Most natural language processing models require large amounts of classified data to achieve optimal performance. However, LRLs often lack sufficient corpora, which leads to several issues, most notably reduced performance. In contrast, high- resource languages like English, French, or Spanich which have huge datasets, allow for the development of highly accurate and high-performance NLP tasks[6].

1.3.2 Lack of Linguistic Resources and NLP Tools

The lack of linguistic resource and language processing tools is another major challenge. Low-resource languages lack databases, annotated corpora, word analyzers, tokenizers, and pre-trained models. The absence of these resources hinders the development of advanced natural language processing applications[6].

1.3.3 Machine Translation Challenges

Machine translation presents another obstacle to natural language processing (NLP) tasks in low-resource languages. Developing effective translation systems requires massive linguistic datasets. The scarcity of balanced datasets and the limited or nonexistent translation resources result in lower accuracy and quality in machine translation systems. Furthermore, words and phrases may have multiple meanings depending on the context, making accurate translation difficult. Ambiguities in translation, spelling variations, and linguistic shifts also complicate NLP tasks, such as sentiment analysis and machine translation, making it crucial to achieve performance comparable to resource-rich languages. To overcome some of these limitations, multi-source data including text, audio, and images can aid in tone of voice analysis or facial expression analysis, leading to a deeper understanding of expressed emotions and more accurate translations. Such multi-source datasets can provide a larger dataset to address the challenges of resource-limited languages and improve NLP applications[6].

1.3.4 Linguistic Complexity

Many low-resource languages are characterized by complex grammatical structures, diverse vocabularies, and rich morphology. These linguistic features make developing natural language processing applications for low-resource languages a significant challenge. Even the diversity of writing systems and scripts further complicates text processing. Moreover, the lack of comprehensive linguistic studies and annotated data hinders the modeling of low-resource languages[6].

1.3.5 Orthographic Variation and Code-Switching

Code-switching and spelling variation are common in languages and dialects with low resource. The same word can be written in multiple ways, especially on social media platforms. Furthermore, most speakers mix different languages within a single sentence, further complicating natural language processing (NLP) tasks, as models must handle multiple linguistic systems simultaneously. These challenges are particularly prevalent in dialects[6], including the Algerian Arabic.

1.4 The Algerian Dialect

The Algerian dialect, known locally as Darija, is the main means of daily communication among Algerian people. It is mainly derived from Arabic, but it has been influenced by several languages, such as Berber, French, Turkish, Spanish, and Italian, due to the cultural and historical interactions that Algeria has experienced over time.

Although the Algerian dialect shares similarities with other Maghrebi dialects, such as Moroccan and Tunisian Arabic, it has its own distinctive linguistic features in terms

of vocabulary, pronunciation, and grammar. It also varies across different geographical regions in Algeria, such as Algiers, Oran, Annaba, and the southern areas. However, despite these regional differences, Algerians generally use a common form of the dialect, especially in social media communication [7].

Unlike Modern Standard Arabic, which is used in education, administration, and official media, the Algerian dialect is mainly used in informal contexts such as daily conversations, songs, homes, and streets [9].

The lack of linguistic resources is one of the main challenges in processing Arabic dialects. Another difficulty is the widespread use of non-standard writing systems, especially Latin script (known as Arabizi) [8].

1.4.1 Characteristics of the Algerian Dialect

Algerian Arabic is characterized by a lack of standardization and limited writing resources, which makes it a low-resource language. It differs from Modern Standard Arabic in several linguistic aspects, including morphology, syntax, lexicon, and phonology. In this section, we present its main characteristics[9].

1.4.1.1 Orthographic Characteristics of the Algerian Dialect

One of the main characteristics of the Algerian dialect is the absence of a unified writing system. As a result, the same word may be written in different ways depending on the writer's preference and language level. This variation is also influenced by the use of different writing systems, such as the Arabic script, Latin script, and Arabizi. Therefore, spelling variations are very common in digital communication [10].

The Algerian dialect uses the same basic vowels and consonants as Modern Standard Arabic. It has also been influenced by several languages due to historical contact with different colonial and governing groups in Algeria.

The Algerian dialect includes additional letters such as ڤ ,ڤ, and پ, which are pronounced as /p/, /g/, and /v/. It also contains many lexical borrowings from different languages. For example, the word سلم (sallam) comes from Modern Standard Arabic. Other examples include فرملي (farmliy) meaning "nurse," borrowed from French; بابور (baabur) meaning "boat," borrowed from Turkish; شلاغم (shlaaghm) meaning "mustaches," borrowed from Berber; زبلة (zabla) meaning "garbage," borrowed from Italian; and سيمانة (siimaanah) meaning "week," borrowed from Spanish [11].

The same word may appear in different forms in a dataset. Therefore, spelling variation remains a major challenge for Natural Language Processing applications.

1.4.1.2 Morphological Characteristics of the Algerian dialect

The Algerian Arabic dialect shows several morphological features that differ from Classical Arabic and are closer to other Maghrebi dialects. These features mainly include simplified conjugation patterns and the use of additional pronouns. In terms of conjugation, the dialect, like other Arabic dialects, often drops case endings in nouns and verbs, while the declarative form is more commonly used. It also tends to lose dual and feminine plural forms, which are usually merged into masculine plural forms. In addition, the dialect includes several morphological phenomena such as suffixed pronouns, simplified verb conjugation, negation using "ma...š," and specific plural formation patterns. Furthermore, it changes some verb structures, especially in the imperfect tense, and introduces dialectal suffixes and phonological variations that distinguish it from Classical Arabic and other Arabic dialects[7].

1.4.1.3 Lexical Variations Characteristics

The vocabulary of the Algerian dialect reflects the influence of several languages that have interacted with Algerian society throughout history. Besides its Arabic foundation, Darija contains numerous borrowed words from Berber, Turkish, French, Spanish, and Italian. For example, words related to daily life, transportation, education, and technology are often borrowed or adapted from foreign languages [7].

This lexical diversity enriches the dialect but introduces additional challenges for NLP systems, particularly in tasks such as sentiment analysis, machine translation, and text classification, where borrowed words and multilingual expressions are frequently encountered.

Algerian	**Translation**	**MSA**	**Origin**
فكرون	a tortoise	سلحفاة	Berber
شلاغم	Moustache	الشارب	Berber
قرجومة	a throat	الحنجرة	Berber
فيشطة	Holiday	عطلة	Italian
زبلة	Garbage	القمامة	Italian
صوارد	Money	النقود	Italian
سيمانة	Week	اسبوع	Spanish
سبيردينة	Snickers	حذاء	Spanish
شكولة	School	مدرسة	Spanish
طابلة	Table	طاولة	French
تيليفون	Phone	هاتف	French
فرملي	Nurse	ممرض	French

Table 1: Examples of loanwords in the Algerian dialect.

1.5 Why is Algerian Dialect a Low-Resource Language

The Algerian dialect is considered a low-resource language due to **data scarcity**, although it's spoken by millions of people. There are a few annotated databases available. This indicates that Darija lacks large annotated corpora, lexical databases, and pretrained language models, unlike high-resource languages such as English and French. Consequently, the development of NLP applications for the Algerian dialect remains limited, making it hard to train machine learning (ML) models [12].

Another important factor is the absence of a standardized writing system(**Standard Orthography**); thus, Algerian people frequently use different writing systems, such as Arabic script, Latin script, and Arabizi, resulting in different spellings for the same word. This variability increases data annotation and the complexity of text processing.

Moreover, the dialect is characterized by extensive **code-switching** between Arabic, French, Berber, and English as a single sentence, for example: rani bien. Additionally, Algerian dialect written in Latin script with numbers that represent Arabic letters(**ARABIZI**), for instance: 3=ع, which is complex for a machine to understand [13].

Furthermore, linguistic mixing creates additional challenges for NLP systems, particularly in text processing, machine translation, dataset creation, and sentiment analysis. such as **regional variation**, **French and Barber influence**, and also Algerians use **sarcasm**, indirect computers fail to recognize the real meaning, for example: "Bravo 3LIK " can mean "you did a terrible" [12].

In addition, **regional variation** leads to the low-resource nature of the dialect. It's complicated to build representative databases and robust language models because of the differences in vocabulary and expressions to convey the same meaning used across different regions [12, 13].

Algerian dialect remains a language with limited resources despite its widespread use in daily communications and social media. Therefore, research and development of resources to support advanced NLP applications are necessary.

1.6 Conclusion

In this chapter, we introduced the concept of Low-Resource Languages and explained why they present significant challenges for Natural Language Processing. We discussed the main characteristics of these languages, including the lack of annotated datasets, limited linguistic resources, and the absence of powerful NLP tools. We also explored several examples of low-resource languages from different regions of the world to show that resource availability is not determined by the number of speakers but by the availability of computational resources.

Then, we focused on the Algerian dialect as the main language of our study. We presented its linguistic characteristics, such as lexical diversity, regional variation, orthographic inconsistency, and the influence of several foreign languages. We also explained why the Algerian dialect is considered a low-resource language by highlighting challenges such as data scarcity, code-switching, Arabizi writing, and the absence of a standardized writing system. These characteristics make many NLP tasks, including sentiment analysis and text classification, more difficult.

Understanding these challenges is essential before developing effective NLP solutions for the Algerian dialect. Therefore, in the next chapter, we will present the fundamental techniques and models used to process textual data. We will begin by introducing word embedding methods, then discuss text classification models, pretrained language models, and deep learning approaches that form the basis of our proposed sentiment analysis system.

Chapter 2

Text Representation and Classification Methods

Introduction

Text representation is a fundamental step in Natural Language Processing (NLP), as machine learning and deep learning models cannot directly process raw textual data. Therefore, text must be transformed into numerical representations that preserve meaningful linguistic information. Various techniques have been developed for this purpose, ranging from traditional statistical methods such as TF-IDF to word embedding approaches and transformer-based models. Recent advances in deep learning have further improved the ability of models to capture semantic and contextual information, leading to significant improvements in text classification and sentiment analysis tasks. This chapter presents the main text representation techniques, classification algorithms, deep learning models, and pretrained transformer architectures used in this work.

2.1 Natural Language Processing

Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) that focuses on enabling computers to understand, interpret, generate, and process human language in a meaningful way. It combines techniques from computer science, linguistics, and machine learning to analyze both written and spoken language. The main objective of NLP is to bridge the gap between human communication and computer understanding, allowing machines to extract information and perform language-related tasks automatically [18].

NLP is widely used in many real-world applications, including sentiment analysis, machine translation, text classification, question answering, speech recognition, spam detection, and chatbots. Over the years, NLP has evolved from rule-based systems to statistical methods, machine learning algorithms, deep learning models, and, more recently, transformer-based pretrained language models. These advances have significantly improved the ability of NLP systems to capture semantic and contextual information from text, leading to higher performance across a wide range of applications [18,19].

2.2 traditional-text-representation-methods

2.2.1 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical approach to represent the data in Natural Language Processing (NLP) that takes into account both Term Frequency (TF) and the Inverse Document Frequency (IDF) for the transformation of text data into numerical feature vectors [12]. TF-IDF is popularly used to assess the importance of terms in documents and is a good representation of textual information for classification and sentiment analysis tasks [16].

Frequency of Term (TF)

The Term Frequency (TF) counts the number of times a term occurs in a document. It is based on the assumption that terms that occur more often are more representative of the content of the document [15]. Therefore, terms with higher frequencies are given more importance in the document representation [16].

$$\mathrm{TF}(t,d) = f(t,d) \tag{2.1}$$

Inverse Document Frequency (IDF)

Is a measure of how rare or common a term is in the whole corpus . The weights of common terms that appear in many documents will be low, and the weights of rare and more informative terms will be high [12]. IDF is calculated from the total number of documents in the corpus and the document frequency of the term, allowing the model to downplay the impact of words that appear frequently but are not very informative [16].

$$\mathrm{IDF}(t) = \log(df(t)N) \tag{2.2}$$

TF-IDF Weighting :

weight is calculated as the product of TF and IDF which can efficiently weight the terms and improve the representation of text data . The term score in TF-IDF is calculated as [16]:

$$\text{TF-IDF}(t,d) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t) \tag{2.3}$$

However, it has many limitations such as being unable to capture semantic relationships between words, lacking contextual understanding and generating high-dimensional sparse vectors[16].

2.2.2 Character TF-IDF

Character TF-IDF represents text using sequences of characters instead of complete words, through character n-grams, which split words into smaller overlapping

character sequences [12]. This representation captures subword patterns rather than relying on whole-word units, making it effective even when words appear in different forms. It is particularly suitable for handling noisy and informal text, especially in social media data where spelling variations and inconsistencies are common. One of its main strengths is robustness to spelling variation, since it can capture similarities between differently spelled words that share similar character structures[15].

This is particularly useful for the case of Algerian Darija and Arabizi, because there is no standard way of writing them and the same word can be written in several ways with either Latin or Arabic characters[12].

2.2.3 Hybrid TF-IDF of Words and Characters

Hybrid TF-IDF is a combination of word-level and character-level representations in one feature space to benefit from both methods[14].

Word-level TF-IDF captures lexical and semantic information from full words, while character-level TF-IDF captures subword patterns and spelling variations. By combining both levels the representation is more balanced by combining lexical meaning and robustness to writing variations[12].

This hybrid approach is especially efficient for Algerian Darija and Arabizi as it retains the semantic information and at the same time manages the inconsistencies of informal and dialectal text[14].

Therefore, hybrid TF-IDF is able to perform more stably and robustly than only using word-level or character-level representations[12].

2.3 Word Embedding Methods

2.3.1 Word2Vec

Word2Vec is one of the most popular word embedding models in the Natural Language Processing (NLP). Proposed by Mikolov et al. in 2013, its goal is to derive dense, low-dimensional vector representations of words from large textual corpora, providing the ability to capture semantic and syntactic relations between words. Unlike traditional sparse representations, Word2Vec maps words into a continuous vector space, where linguistically similar words are located close to each other. This property enables the model to learn meaningful relationships and analogies between words.

$$\text{e.g., } king - man + woman \approx queen \text{ [18]}$$

The Word2Vec model uses two main architectures for the learning process: Continuous Bag-of-Words (CBOW) and Skip-Gram. These architectures differ by how they leverage contextual information for learning vector representations. Both methods aim to model the semantics of the words, but they use opposite prediction strategies

during training [17].

Continuous Bag-of-Words (CBOW) :

The CBOW architecture predicts a target word from the surrounding context words. In particular, the model is given surrounding words as input and asked to predict the most probable word that would fit in the given context. CBOW is very effective for using frequent words and large datasets as it is computationally more efficient and trains faster [17].

Skip-Gram :

In contrast, the Skip-Gram architecture takes a target word and predicts the context words around it [17]. Skip-Gram learns more informative semantic representations by modeling the relationship between a word and its neighboring terms and is particularly effective for capturing relationships involving rare or infrequent words [18]. Although it generally takes longer to train than CBOW, it usually yields higher quality embeddings for complex linguistic patterns [17].

Word2Vec, despite being efficient and scalable, produces static embeddings, assigning a single vector to each word regardless of its contextual usage. Therefore, the words with multiple meanings are represented the same in all contexts, which limits the model's ability to handle polysemy. These limitations motivated the development of more advanced contextualized embedding models such as ELMo and BERT that produce context-dependent word representations [18].

2.3.2 FastText:

FastText is an extension of the Word2Vec model that enhances the Skip-gram architecture by taking into account the internal structure of words, rather than treating words as independent units [13]. The model employs subword information during the training stage, which enables it to utilize both the morphological structure of words and their context [18]. Furthermore, each word is represented as a set of character n-grams instead of a single token. This approach results in more expressive vector representations and allows the model to more efficiently capture the linguistic and structural relationships between words [19].

Subword Embeddings is based on the idea where the representation of a word is a composition of the embeddings of its subword units, instead of the whole word alone [19]. These subword components are represented and combined to obtain the final word vector [13]. This mechanism enables the model to better produce meaningful embeddings for rare words and Out-of-Vocabulary (OOV) words. It also helps to capture morphological patterns and linguistic relationships shared among different words[18].

n-grams of characters :

FastText is based on the use of Character n-grams which are sequences of consecu-

tive characters that are used as the basic learning units [18]. Each word is represented using a set of character n-grams, so that the model can learn morphological and structural information contained in words [19]. For example, the word unhappiness can be broken into subword units un, happy and ness, allowing the model to learn common morphological patterns [18]. Similarly, in Algerian Darija, the word عجبني (I liked it) in the sentence "الساك عجبني بزاف" (I really liked the bag) can be segmented into character n-grams such as عج, عجب, جبن and بني . This decomposition allows the model to exploit internal structure of words and produce useful representations for rare or out-of-vocabulary words[13].

Relevance to Misspellings and Dialect Words:

A major advantage of FastText is its ability to handle misspellings and dialectal variations, thanks to its subword based representation. The model can use shared character sequences to generate similar vector representations even if words are written differently. This ability is especially useful for dealing with Out-of-Vocabulary (OOV) words, which are often found in informal texts and social media [13].

Benefits of Using FastText for Algerian Arabic :

The high linguistic diversity and the large variety of spelling variations in user-generated content make FastText particularly well suited for dealing with the Algerian Darija. Using Character n-grams and Subword Embeddings, the model can handle words written in different orthographic forms such as Arabizi and non-standard spellings. FastText helps to improve the performance in Natural Language Processing (NLP) applications such as Sentiment Analysis, Text Classification and semantic information extraction from informal Algerian textual data [13].

2.4 document-representation-methods

doc2vec

The model Doc2Vec is based on a central concept in distributional semantics, which states that the meaning of a word is given by the contexts in which it occurs in texts [22]. To extend this idea from the level of words to the level of entire texts, Doc2Vec was developed as an extension of the Word2Vec model, with the aim of learning vector representations for paragraphs and documents in addition to words . During the learning process, the model takes into account the words present in the text and the document identifier itself, which helps it capture the semantic and contextual information that distinguishes each document [20]. This model was introduced in two main versions: Distributed Memory (DM) and Distributed Bag of Words (DBOW), both inspired by the CBOW and Skip-Gram algorithms used in Word2Vec [21]. Thanks to its ability to represent the overall meaning of a text in a low-dimensional numerical vector, Doc2Vec has become one of the methods used in many Natural Language Processing applications, such as text classification, sentiment analysis, and information

retrieval.

2.5 Classification Models

2.5.1 Machine Learning

Machine learning (ML) is a branch of Artificial Intelligence (AI) that focuses on developing models capable of learning patterns from data and making predictions or decisions without being explicitly programmed for every task. The term *machine learning* was introduced by Arthur Samuel in 1959, who defined it as the ability of computers to learn without being explicitly programmed. Instead of relying on manually designed rules, machine learning algorithms improve their performance by learning from examples contained in training datasets. These learned patterns enable the models to generalize to unseen data and make accurate predictions. Machine learning has become one of the most widely used approaches in Natural Language Processing (NLP), where it is applied to tasks such as text classification, sentiment analysis, spam detection, and document categorization [21].

2.5.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm widely used for classification tasks. It operates by constructing an optimal hyperplane that separates data points belonging to different classes while maximizing the margin between them. The data points closest to the decision boundary are known as support vectors, and they play a crucial role in defining the classification model. Due to its margin maximization principle, SVM demonstrates strong generalization capabilities and is particularly effective for high-dimensional and sparse datasets, such as text data [21, 23].

One of the main advantages of SVM is its ability to handle both linearly and non-linearly separable data through the use of kernel functions. Linear kernels are commonly employed in text classification because textual representations often involve large feature spaces. In sentiment analysis, SVM has shown strong performance in distinguishing positive and negative sentiments and has achieved high classification accuracy across different datasets [21]. However, its performance may decrease when dealing with neutral sentiment due to the overlap between sentiment categories. Furthermore, SVM can be sensitive to overfitting and class ambiguity, which should be considered when applying it to sentiment classification tasks [23, 24].Figure 2.1 illustrates how Support Vector Machine (SVM) separates different classes by constructing decision boundaries (hyperplanes) that maximize the margin between them.

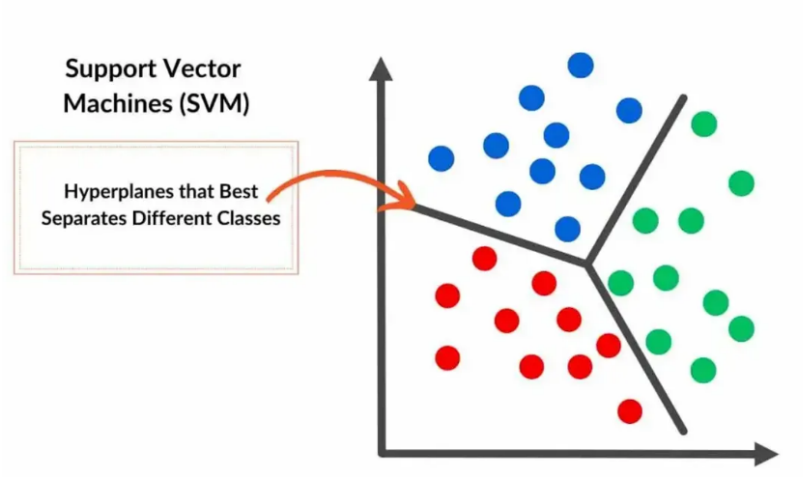


Figure 2.1: Illustration of the Support Vector Machine (SVM) decision boundaries separating different classes.

2.5.3 Logistic Regression

Logistic Regression is a supervised probabilistic classification algorithm designed to estimate the probability that an instance belongs to a particular class. Unlike linear regression, which produces continuous outputs, Logistic Regression employs the sigmoid function to constrain predictions to values between 0 and 1, making it suitable for classification problems. This probabilistic framework allows the model to assign instances to classes based on estimated probabilities [24, 21].

Logistic Regression is known for its simplicity, computational efficiency, and ease of interpretation. It performs well with sparse textual features and is commonly used in sentiment analysis tasks. The model generally provides stable performance and good generalization ability, particularly when the data exhibits approximately linear separability [23]. Nevertheless, its linear decision boundary limits its capacity to capture complex non-linear relationships between features. Similar to SVM, Logistic Regression performs effectively in identifying positive and negative sentiments but may encounter difficulties in classifying neutral sentiment because of lexical and contextual overlap between classes. Despite these limitations, it remains one of the most widely used algorithms for text classification and sentiment analysis [23].

2.6 Deep learning

Is a core component of artificial intelligence (AI) and machine learning (ML) that mimics how the human brain processes data. It relies on computational models built with multiple layer structures, known as artificial neural networks (ANNs), to automatically extract and analyze complex features from large amounts of information. Unlike traditional machine learning techniques, which often require manual step-by-step feature engineering, deep learning excels at learning directly from raw his-

torical data. Because of this powerful data-driven learning capability, deep learning is widely used to build automated and intelligent software across various fields, including cybersecurity, healthcare, and smart computing[25].

2.6.1 Bidirectional Long-Term Memory (BiLSTM)

Bidirectional long-term memory (BiLSTM) is an extension of the Long-Term Memory Network (LSTM), processing sequential data in both forward and backward directions. Unlike standard LSTM, which only considers forward information, BiLSTM incorporates both forward and backward contextual information, allowing for a more comprehensive representation of textual data [26].This architecture consists of two LSTM layers: a forward layer that processes the sequence from left to right and a backward layer that processes it from right to left. The outputs of these two layers are combined to generate context-aware representations, improving the understanding of semantic relationships within the text [25, 27]. Figure 2.2 illustrates the architecture of a Bidirectional Long Short-Term Memory (BiLSTM) network, where two LSTM layers process the input sequence in opposite directions and their outputs are combined to capture contextual information from both the past and the future.

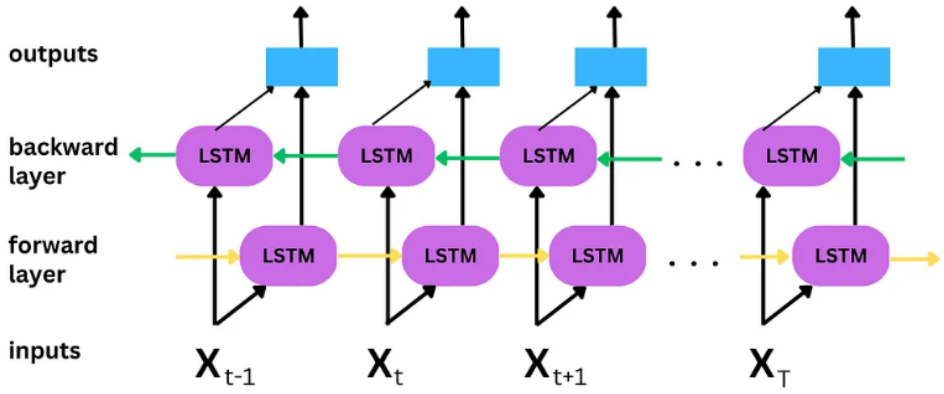


Figure 2.2: Architecture of the Bidirectional Long Short-Term Memory (BiLSTM) network.

Because of its ability to accommodate long-term dependencies and contextual information from both directions, BiLSTM has been widely used in many natural language processing tasks, including text classification, sentiment analysis, named entity recognition, and machine translation [27].

2.6.2 Transformer Encoder

Transformer Encoder Overview

The main purpose of the Transformer encoder is to change a sequence of input tokens into a continuous sequence of vector representations [29]. By using only attention mechanisms instead of traditional recurrent or convolutional layers, it removes the need for sequential step-by-step computation and processes all token positions at the same

time. This design is important because it allows for high parallelization during training while handling long-range connections between words very efficiently [30]. Figure 2.3 illustrates the architecture of a Transformer encoder layer. Each encoder layer consists of a Multi-Head Self-Attention module followed by a position-wise Feed-Forward Network. Both sub-layers are surrounded by residual connections and layer normalization, and the encoder block is repeated N times to build the complete encoder.

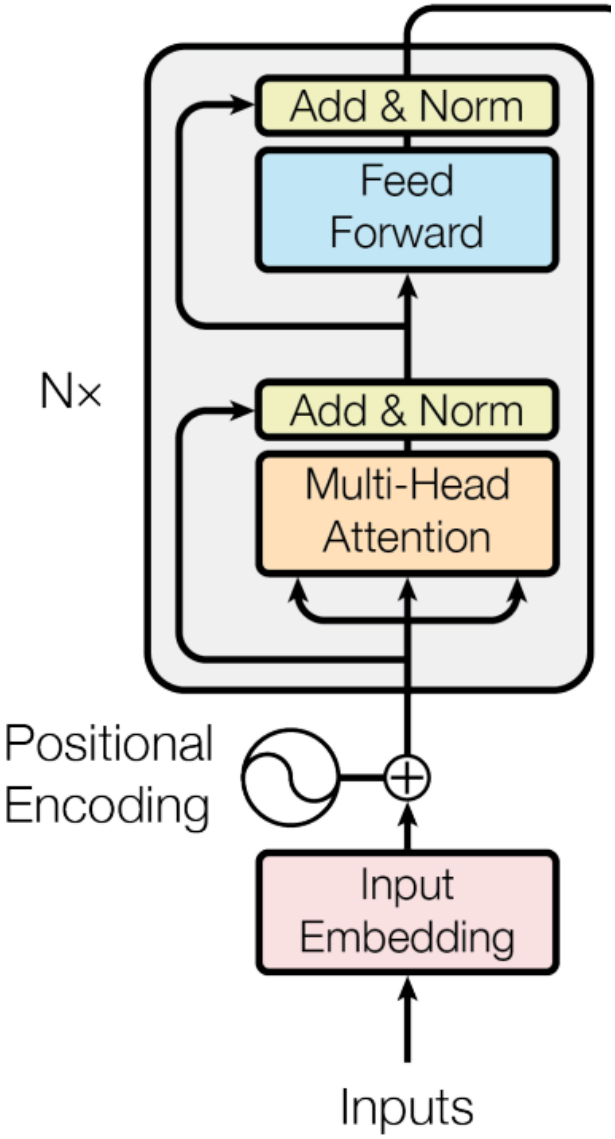


Figure 2.3: Architecture of the Transformer encoder.

Self-Attention

The self-attention mechanism updates the representation of a sequence by looking at how different words relate to each other within the same sentence [29,30]. It relies on Scaled Dot-Product Attention, which takes queries (Q), keys (K), and values (V), scales their dot products by the key dimension size ($\sqrt{d_k}$), and applies a softmax

function to distribute attention weights [29]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.4}$$

This mechanism is crucial because it helps the encoder dynamically understand context and catch dependencies across long text distances, avoiding the vanishing gradient issues found in older recurrent networks [29,30].

Multi-Head Attention

Multi-head attention extends basic self-attention by splitting the queries, keys, and values into multiple smaller, learned linear projections that run in parallel [29]. This setup allows the model to look at information from different representation subspaces and positions at the same time [29,30]. It is important because it prevents the loss of specific details that happens when averaging everything into a single attention view, improving the model's ability to catch complex grammar and meaning [29].

Feed-Forward Network

Inside every encoder layer, there is a position-wise feed-forward network (FFN) that processes each token position separately and in the exact same way. This network uses two linear transformations with an activation function in between, adding necessary non-linearity to the model [29]. The FFN is essential because it boosts the model's capacity to process and refine the contextual features built by the attention layers [29,30].

Residual Connections

To help gradients flow smoothly through the deep architecture, every sub-layer in the encoder—including the attention block and the FFN—is wrapped in a residual connection [29]. This design adds the original input directly to the output of the sub-layer function. These shortcut paths are important because they prevent gradients from vanishing or exploding during training, which makes it much easier to train deep and highly parameterized Transformer networks [29,30].

Layer Normalization

Layer normalization is used throughout the encoder to keep the distribution of layer activations stable [29,31]. Instead of normalizing across a whole batch of data, it calculates the mean and variance statistics across all the hidden units within a single layer for each individual training example [31]. This approach is vital because it removes dependencies between different training samples and keeps the scale of forward and backward signals steady, reducing training instability and cutting down overall training time [29,31].

2.7 Pretrained Transformer Models

Recent advancements in Natural Language Processing (NLP) have been driven by the adoption of large-scale, pretrained transformer-based architectures. These models leverage self-attention mechanisms to generate deep contextualized text representations, establishing new performance benchmarks across numerous downstream applications [32,33]. While early models predominantly catered to high-resource languages like English, recent research has shifted toward developing language-specific and dialect-specific transformers to address the unique morphological and syntactic features of other tongues [33,34].

2.7.1 AraBERT

AraBERT was introduced as a pioneer monolithic bidirectional transformer model trained specifically for the Arabic language to address the limitations of multilingual models, which often underperform on highly inflected and morphologically rich languages. Developed using the standard $\text{BERT}_{\text{Base}}$ configuration, the model was pretrained on a comprehensive corpus consisting of roughly 70 million sentences (about 24 GB of text) aggregated from heterogeneous sources, including news articles and Wikipedia. A core distinction of AraBERT is its advanced text preprocessing pipeline, which segments prefixes and suffixes to overcome vocabulary sparsity, thereby achieving state-of-the-art results across fundamental Arabic language tasks such as Sentiment Analysis (SA) and Named Entity Recognition (NER)[32].

2.7.2 MARBERT

To bridge the gap between Modern Standard Arabic (MSA) and the conversational language used on social networks, MARBERT (Multi-dialectal Arabic BERT) was engineered as a massive multi-dialectal model [46]. Unlike models trained predominantly on formal text, MARBERT was pre-trained on a massive social media dataset encompassing over 1 billion Arabic tweets containing roughly 15.6 billion tokens [33]. This specialized training enables the model to effectively process both MSA and regional dialects. When evaluated on complex benchmarks, MARBERT demonstrated an exceptional capacity for handling non-standard syntax, informal language variations, and social media classification tasks, making it highly effective for dialectal sentiment analysis [33,34].

2.7.3 DziriBERT

While multi-dialectal models cover broad regional variations, low-resource localized dialects require dedicated architectures. DziriBERT was developed as the first autonomous language model custom-tailored for the Algerian dialect (Darja). The Algerian speech register is highly unique due to its integration of standard Arabic with

words from French and Berber languages, alongside its frequent transcription in Roman characters (Arabizi). To capture these specificities, DziriBERT was trained on a localized corpus of 1 million Algerian tweets (around 150 MB of text data) using a Masked Language Modeling (MLM) probability of 25%. Despite its relatively small pre-training data volume, DziriBERT outperforms massive multilingual and regional Arabic transformers, particularly when classifying informal scripts and Romanized text formats[33].

2.7.4 XLM-RoBERTa

Is a multilingual language model based on the Masked Language Model (MLM), developed by the Facebook AI team to support large-scale cross-linguistic learning . This model was trained to support 100 different languages simultaneously, using a massive amount of filtered data from the Common Crawl project, exceeding 2 terabytes . XLM-R is highly efficient in handling languages with limited resources, enabling the transfer of linguistic knowledge between different languages without negatively impacting the performance of each individual language. It also significantly outperforms the mBERT model in classification and named entity recognition tests [35].

Conclusion

This chapter provided an overview of the main techniques used to transform textual data into numerical representations suitable for machine learning and deep learning models. It covered traditional methods such as TF-IDF, as well as modern approaches including word embeddings like Word2Vec, FastText, and Doc2Vec. It also introduced classical classifiers such as Logistic Regression and Support Vector Machines, along with deep learning architectures like BiLSTM and Transformers, and finally presented several pretrained language models used for Arabic and dialectal text processing. The next chapter will focus on Sentiment Analysis as the main application of these techniques.

Chapter 3

Sentiment Analysis

Introduction

Sentiment analysis has become a key research area in Natural Language Processing (NLP) because it can automatically extract opinions, emotions, and attitudes from text. The massive growth of user content on social media, online forums, and review sites has driven research in this field, opening up new ways to analyze public opinion using computational tools. Even though sentiment analysis has advanced significantly, the task is still very difficult for Arabic dialects, especially the Algerian dialect.

This chapter presents the theoretical foundations of sentiment analysis. It first introduces the fundamental concepts related to sentiment analysis, including opinions, emotions, and subjectivity. Next, it presents the different levels and categories of sentiment classification. After that, it discusses the main challenges of analyzing sentiment in the Algerian dialect and reviews the most relevant research on Arabic and Algerian dialect sentiment analysis.

3.1 Fundamentals of Sentiment Analysis

3.1.1 Definition of Sentiment Analysis

Sentiment analysis (SA), also called opinion mining, is an NLP technique that detects, extracts, and classifies opinions, sentiments, and attitudes concerning different topics of various types of data, such as text, video, and audio [36]. The purpose of sentiment analysis is to automatically determine whether a text is positive, negative, or neutral.

3.1.2 Opinion, Emotion, and Subjectivity

In sentiment analysis, the concepts of opinion, emotion, and subjectivity are closely related but focus on different sides of human expression. Recognizing the differences between them is crucial since they each affect how text data is interpreted. Generally, an opinion is an evaluation or judgment about a specific entity or feature. An emotion shows an internal feeling, like happiness or anger. Finally, subjectivity covers personal beliefs, views, and attitudes [36]. These concepts form the foundation of sentiment analysis and are discussed in the following sections.

1. **Opinion.**

An opinion is defined as a personal expression that shows a positive or negative bias, evaluation, or attitude toward a specific entity, event, or its characteristics. In the context of sentiment analysis, opinions are studied structurally as emotional expressions passed from an opinion holder to a target audience [37].

These opinions can take many different forms depending on their target and how they are written. In sentiment analysis, detecting opinions is crucial because they serve as the main source of sentiment information found within text data [38].

2. **Emotions.**

   Emotions are a set of psychological and mental states that can be described as subjective internal sensations, such as love, joy, surprise, anger, sadness, and fear. These six patterns constitute the primary affective pillars in psychological and computational literature [37].

   It is important to separate emotions and opinions. Even though people often mix them up, emotions are internal affective states experienced by an individual, like being happy or angry. On the other hand, opinions are evaluative judgments and cognitive attitudes projected onto a target entity or a specific aspect of it [38].

3. **Subjectivity.**

   Subjectivity is the linguistic expression of personal feelings, internal beliefs, and mental states, as opposed to the presentation of verifiable factual data. The scope of subjective language is broad, including not only opinions but also claims, suspicions, assumptions, and desires [37].

   To see how broad this is, consider the following example: "I wanted a phone with good sound quality." This sentence is subjective in essence, expressing an internal personal desire, but it lacks the characteristics of an opinion because it does not directly evaluate any specific actual phone. This difference proves that subjectivity is a holistic linguistic concept that includes opinions within its various branches without being limited to them [38].

3.1.3 Objective and Subjective Texts

For computational text processing, sentiment analysis frameworks often classify linguistic data into objective and subjective data based on whether the text expresses abstract facts and personal preferences [37].

Objective text:

"This past Saturday, I bought a Nokia phone, and my girlfriend bought a Motorola phone."

"We called each other when we got home."

Both sentences deal with verifiable facts and factual actions, and are therefore considered objective [38].

Subjective Text:

Subjective sentences move beyond simply stating facts to include expressing impressionistic judgments and personal assessments. For example:

"The voice on my phone was not so clear, worse than my previous phone."

This statement is subjective in nature because it is based on a personal, comparative evaluation of the device's quality [38].

3.1.4 Explicit and Implicit Opinions

Classifying opinions according to their structure (explicit or implicit) is one of the major challenges in computational linguistics.

Explicit opinions: These are statements in which the intended feeling and evaluative polarity are clearly expressed through explicit affective modifiers.

Example: "The voice quality of this phone is amazing."

Implicit opinions: These are statements that lack direct affective indicators but are inferred from facts or context.

Example: "The earphones broke within two days." [38]

3.2 Levels of Sentiment Analysis

3.2.1 Document-Level SA

At this level, the whole document is treated as a single information unit. The objective is to classify the sentiment of an entire document as either positive, negative, or neutral [39]. However, this approach fails with mixed reviews, for example: "I love the food, but I hate the service."

3.2.2 Sentence-Level SA

In this approach, the aim is to classify the sentiment expressed in each individual sentence. First, it determines whether the sentence is subjective (expresses an opinion) or objective (states a fact). If a sentence is subjective, it classifies whether it is positive, negative, or neutral [39].

3.2.3 Aspect-Level SA

It classifies sentiment with respect to specific aspects or features of an entity. Then, it determines the sentiments for each aspect. This level of analysis becomes important because classification at the document or sentence level does not provide the necessary detail [5]. **Example context: "A Restaurant Review."**

"The pasta was absolutely delicious. However, the waiter was rude, and the service

was incredibly slow."

Entity	Aspect	Sentiment
The restaurant	food(pasta)	Positive
	service(waiter)	Negative
	service(speed)	Negative

Table 3.1: Example of Aspect-Level Sentiment Analysis

3.3 Sentiment Classification Categories

Sentiment analysis and opinion mining systems categorize computational data into different levels of granularity depending on the specific objectives of the natural language processing (NLP) task [40]. Sentiment classification is generally structured into three prominent operational categories: binary, multi-class, and fine-grained classification.

3.3.1 Binary Classification

Binary classification represents the most foundational approach to sentiment analysis, wherein textual data is classified into two categories: Positive and Negative [41], [40]. This approach focuses on determining the overall emotional orientation of the text, intentionally filtering out neutral and objective statements to optimize model performance on extreme polarities [41].

3.3.2 Multi-Class Classification

Multi-class classification (frequently operationalized as three-class classification) expands the binary paradigm by introducing an intermediate dimension to capture non-polarized expressions [40]. Within this framework, textual units are categorized into three distinct classes: Positive, Neutral, and Negative [41,40]. The integration of the neutral class is highly critical in computational linguistics, as it accommodates objective facts, queries, and multi-faceted opinions that do not exhibit a definitive positive or negative inclination [40]. This classification scheme is adopted in our study.

3.3.3 Fine-Grained Classification

Fine-grained sentiment classification provides a deeper, nuanced understanding of text by measuring the intensity and strength of the expressed emotions [40]. This approach typically relies on a five-class classification scheme, which spans across a spectrum of five ordinal categories: Very Positive, Positive, Neutral, Negative, and Very Negative [40]. This approach provides a more detailed representation of emotions by distinguishing between different levels of positive and negative intensity [41,40].

3.4 Challenges of Sentiment Analysis in the Algerian Dialect

Sentiment analysis in the Algerian dialect remains a challenging task due to several linguistic and computational factors. Algerian Arabic exhibits substantial regional diversity, frequent language mixing, non-standard writing practices, and limited linguistic resources. These characteristics complicate text preprocessing, feature extraction, and sentiment classification, making the development of accurate sentiment analysis systems particularly difficult [42,43].

3.4.1 Dialectal Variations

Algerian Arabic is characterized by significant regional diversity, with distinct dialects spoken across different regions of the country. These dialects differ in their lexical, phonological, and morphological characteristics, creating substantial linguistic variability. Such diversity poses challenges for sentiment analysis systems because language patterns learned from one regional variety may not generalize effectively to another [44].

Furthermore, dialectal Arabic differs considerably from Modern Standard Arabic (MSA). The presence of multiple dialectal forms and regional linguistic variations increases the complexity of automatic sentiment classification and requires models to process a wide range of linguistic structures and expressions [42,43].

3.4.2 Code-Switching

Code-switching is one of the most prominent characteristics of Algerian social media content, where users frequently alternate between Arabic and French, and sometimes English, within the same sentence or comment. This linguistic mixture affects grammar, vocabulary, and sentiment indicators, thereby increasing the complexity of sentiment analysis tasks [42,43].

For example, the expression “Les hommes, ربي يوفقك” combines French and Arabic within a single statement. In English, it can be translated as “You are the man, may God help you succeed.” This example illustrates how multilingual expressions require sentiment analysis systems to process multiple languages simultaneously while preserving the intended meaning and sentiment of the text [44].

3.4.3 Arabizi Writing

Arabizi is an informal writing system widely used on social media to represent Arabic words phonetically using Latin letters and numerals. Despite its widespread adoption among users, Arabizi remains highly variable because it lacks standardized writing conventions, making automatic processing particularly challenging.

For example, the numerals 3, 7, and 9 are commonly used to represent the Arabic letters ع, ح, and ق, respectively. Such writing practices increase the difficulty of

normalization, tokenization, and feature extraction, thereby complicating sentiment classification tasks and other natural language processing applications [44].

3.4.4 Orthographic Variations

Orthographic variation represents another important challenge in Algerian dialect sentiment analysis. Algerian Arabic may be written using Arabic script, Latin script, or a combination of both. In addition, the absence of standardized spelling conventions leads to considerable variation in written forms across users and platforms [43,44].

For example, the same word can be written in several different ways, such as "بزاف", "بزااف", or "بززاف", while keeping the exact same meaning. These spelling inconsistencies make the text highly irregular and complicate the preprocessing stages. As a result, sentiment analysis models can struggle to recognize that these are the same words, which can reduce feature representation quality and overall classification performance.

3.4.5 Resource Scarcity

The Algerian dialect remains relatively underrepresented in sentiment analysis research. Existing studies highlight the shortage of linguistic resources and benchmark datasets specifically designed for Algerian Arabic, limiting the development and evaluation of robust sentiment analysis systems [42,43]. This scarcity of publicly available datasets has motivated the creation of new Algerian dialect resources in recent years.

Moreover, the lack of standardized preprocessing tools and benchmark evaluation resources creates additional challenges for building effective sentiment analysis pipelines. As a result, resource scarcity continues to be a major obstacle to advancing research on Algerian dialect sentiment analysis [44].

3.4.6 Sarcasm and Figurative Language

The detection of sarcasm and figurative language is another major challenge in sentiment analysis. Sarcastic expressions often convey sentiments that differ from their literal meaning, making automatic sentiment identification particularly difficult.

For example, the sarcastic expression "واش انت إنسان ولا لتر زيت مقلي؟" was translated in the reference as "Are you a human or a liter of fried oil?" Although the statement appears humorous, it implicitly criticizes a person's appearance or behavior and conveys a negative sentiment. Such expressions may be misclassified by sentiment analysis models that lack sufficient knowledge of dialect-specific language and cultural context [44].

The challenges discussed above illustrate the complexity of sentiment analysis in the Algerian dialect. Linguistic variation, code-switching, Arabizi writing, orthographic inconsistencies, limited resources, and figurative language significantly increase the

difficulty of automatic sentiment classification. These challenges motivate the development of specialized datasets and models tailored to the characteristics of Algerian Arabic.

3.5 Related Work

Research on Algerian dialect sentiment analysis has grown in recent years due to the increasing availability of social media data and the need for language tools built for local dialects. However, compared to English and Modern Standard Arabic, the Algerian dialect still suffers from a lack of available datasets and linguistic resources.

One of the earliest and most important contributions was made by Guellil et al., who built a large corpus of Algerian social media comments for sentiment analysis. This dataset consists of 373,984 YouTube comments and shows the multilingual nature of Algerian online communication, featuring a lot of code-switching between Arabic, French, and English, which accurately reflects the real language patterns found in Algerian social media interactions [45].

Later, Mazari and Kheddar introduced an Algerian sentiment corpus composed of approximately 14,000 comments collected from several social media platforms such as Facebook, YouTube, and Twitter [46]. The dataset includes both Arabic and Latin scripts and addresses important challenges such as Arabizi writing and dialectal variation.

Recently, transformer models have achieved highly promising results for processing the Algerian dialect. Abdaoui et al. developed DziriBERT, a pretrained language model dedicated to the Algerian dialect. The model was trained on 1.1 million Algerian tweets written in both Arabic and Arabizi. Experimental results showed good performance in processing dialectal texts, especially Romanized Algerian Arabic [47].

Study	**Contribution**	**Dataset**	**Technique**
Guellil et al. (2021)	Large Algerian sentiment corpus	373,984 comments	CNN-based SA
Mazari & Kheddar (2022)	Algerian sentiment dataset	~14K comments	ML algorithms, CNN, GRU
Abdaoui et al. (2021)	DziriBERT	1.1M tweets	Transformer
Lanasri et al. (2023)	Public sentiment model	Social media texts	Fine-tuned transformers

Table 3.2: Comparative Table of Related Work

Similarly, Lanasri et al. proposed a sentiment classification framework for the Algerian dialect based on fine-tuned pre-trained models. The system supports both Arabic and Latin scripts and was made publicly available on the Hugging Face platform, providing a useful resource for future research [48].

These advancements show how important it is to build specialized resources for low-resource languages. Ultimately, this work opens the door for more detailed and effective sentiment analysis across different language contexts.

Conclusion

This chapter presented an overview of sentiment analysis and the main concepts required for understanding sentiment classification tasks. It explained the core ideas related to opinions, emotions, and subjectivity, as well as the different levels and categories of sentiment classification. The chapter also discussed the major challenges of sentiment analysis in the Algerian dialect, including dialectal variation, code-switching, Arabizi, orthographic inconsistencies, limited linguistic resources, and figurative language. In addition, a review of previous studies highlighted the progress made in Arabic and Algerian dialect sentiment analysis while emphasizing the challenges that remain due to limited resources and the complexity of dialectal Arabic. The concepts and challenges presented in this chapter provide the theoretical foundation for the methodology and experiments described in the following chapters.

Chapter 4

Methodology

Introduction

This chapter outlines the methodology behind our approach to Algerian dialect sentiment analysis. First, we describe how we built the AlgeriaPulse corpus, covering data collection, annotation, dataset integration, balancing, and preprocessing. Next, we present the text representation and feature extraction techniques applied in this work, including TF-IDF, word embedding models, and fusion methods. After that, we describe the sentiment classification models used in our experiments, including traditional machine learning algorithms, deep learning methods, and pretrained transformer language models. We then introduce the proposed Hard Majority Voting ensemble, in which the predictions of the best-performing models are combined to produce the final sentiment classification. Finally, we present the evaluation metrics used to assess and compare the performance of the proposed approaches.

4.1 Methodological Workflow

The work aims to develop and evaluate sentiment analysis models for the Algerian dialect within a low-resource language context, where the main challenge is the lack of labeled datasets and linguistic tools. To achieve this, we proposed an approach that mixes unsupervised representation learning with supervised sentiment classification. First, we use unlabeled comments to train word embeddings. Then, we use the labeled data to train and test different sentiment models, including classic machine learning, deep learning, and transformers.

First, we collected a large dataset of Algerian social media comments to create the AlgeriaPulse corpus. This dataset is divided into three parts: an unlabeled corpus, a labeled sentiment corpus, and a manually annotated test set.

The unlabeled corpus was used to train domain-specific Word2Vec, FastText, and Doc2Vec embedding models, which were then used as input representations for both machine learning and deep learning classifiers. In addition, several TF-IDF-based representations, including character-level TF-IDF, hybrid, and fusion approaches, were investigated as baseline feature extraction methods. After that, we performed sentiment classification using several different methods. We used traditional machine learning algorithms, such as Logistic Regression (LR) and Support Vector Machines (SVM), as well as deep learning architectures, such as BiLSTM and Transformer Encoder networks. Finally, we evaluated pretrained transformer models, including AraBERT,

MARBERT, DziriBERT, and XLM-RoBERTa. The predictions of the best-performing models, together with Gemini, were then combined using the proposed Hard Majority Voting ensemble to produce the final sentiment classification.

The overall methodological workflow adopted in this study is illustrated in Figure 4.1.

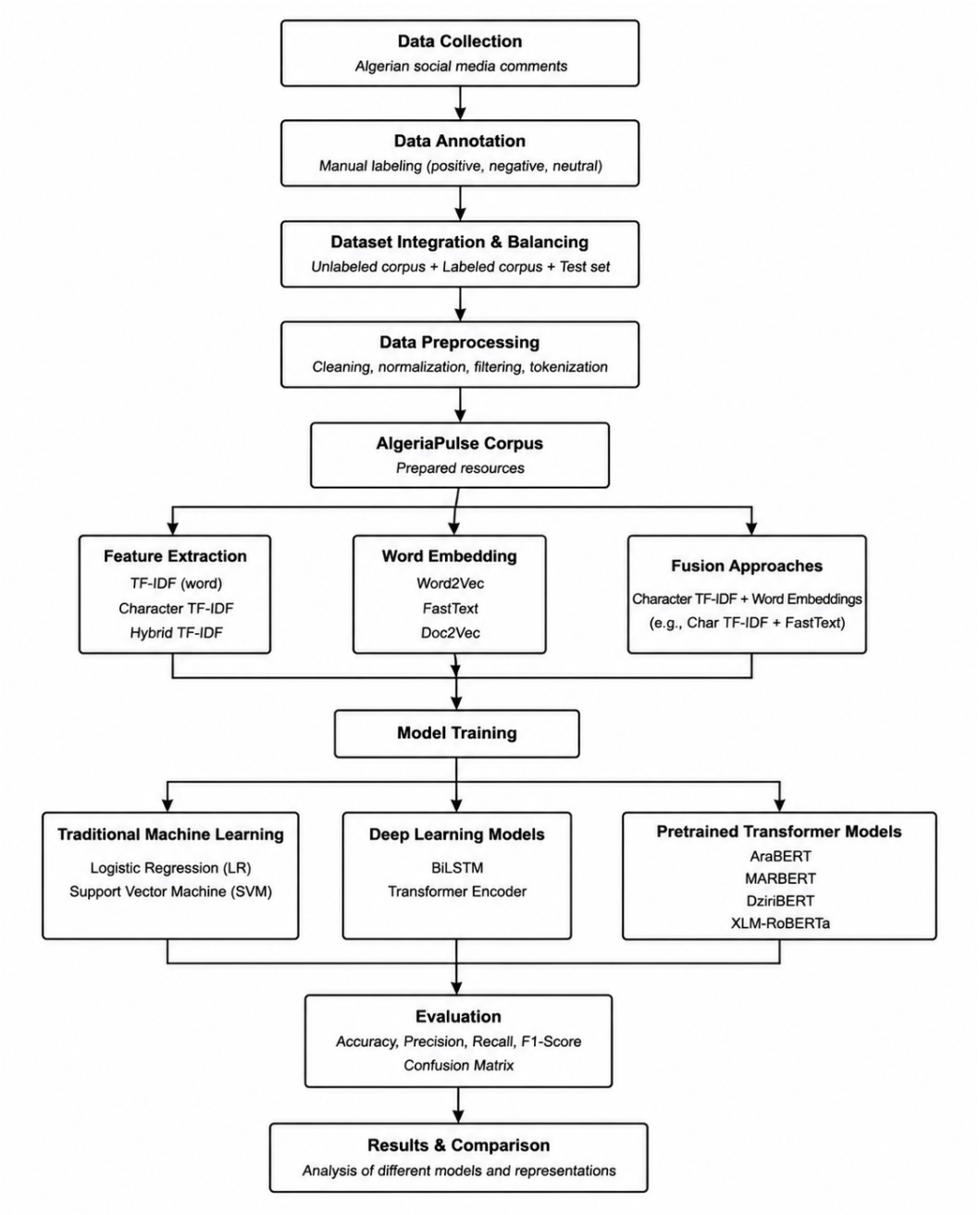


Figure 4.1: Workflow diagram of our proposed approach.

4.2 AlgeriaPulse Dataset Construction

Given that Algerian Arabic is a low-resource language (LRL), it faces several challenges, as discussed in Section 1.5, including data scarcity. We created a new corpus

under the name AlgeriaPulse with the aim of developing and evaluating natural language processing (NLP) tasks.

4.2.1 Data Sources

Data were collected from several social media platforms, primarily YouTube through the YouTube Data API, in addition to Facebook and Instagram. These platforms were selected because they contain a large volume of Algerian dialect content spanning various domains, such as sports, politics, news, entertainment, gaming, cooking, lifestyle, and social discussions. To ensure diversity in topics, writing styles, and user interactions, a large number of Algerian YouTube channels and videos from different categories were selected. Comments were extracted using custom Python scripts and the YouTube Data API.

4.2.2 Corpus Construction

Following the data collection stage, we organized the collected resources into two separate corpora.

First, we built a large unlabeled corpus from the comments gathered from social media platforms. This corpus contains about 1.47 million comments written in the Algerian dialect, and we used it to train the Word2Vec, FastText, and Doc2Vec embedding models.

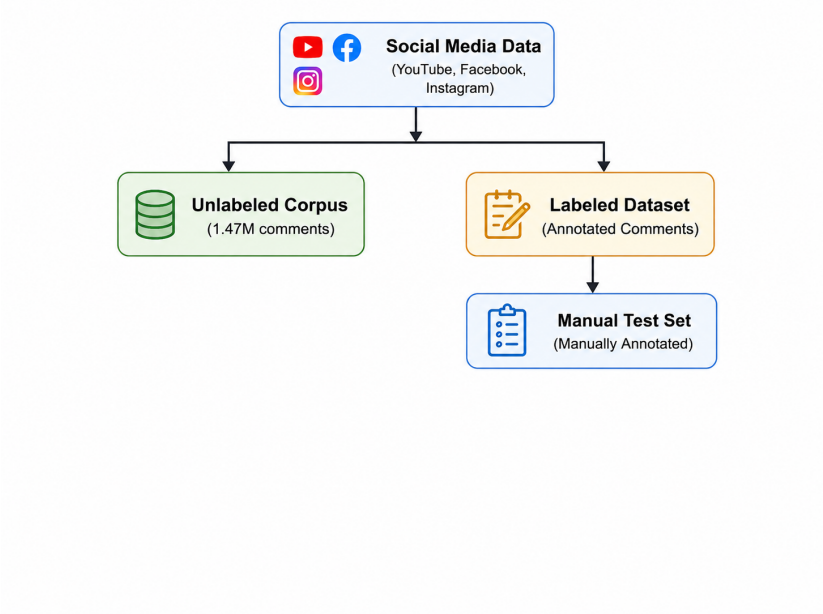


Figure 4.2: Corpus Construction Framework (generated with AI assistance).

Second, we put together a labeled sentiment corpus for our classification task. To do this, we gathered comments collected specifically for sentiment analysis to build our dataset. This gave us a good amount of data and diversity to work with. We then used this final corpus to train and test our different sentiment classification models. Finally,

we selected a small part of the labeled corpus to create an independent test set. We kept this test set exclusively for the final evaluation stage. Figure 4.2 describes our corpus construction framework.

4.2.3 Annotation

The data classification process was performed using Label Studio with Qwen 2.5 and Llama 3.2, which ran locally via Ollama. The annotation process involved sentiment analysis, where each comment was classified as positive, negative, or neutral.

Require: Set of collected comments $C = \{c_1, c_2, \ldots, c_n\}$ ▷ Input comments
Require: Local LLMs $M = \{\text{Qwen 2.5}, \text{Llama 3.2}\}$ (via Ollama) ▷ Models
Require: Label Studio project ▷ Annotation platform
Ensure: Annotated dataset D ▷ Each comment labeled as positive, negative, or neutral

$D \leftarrow \varnothing$ ▷ Initialize empty dataset
Create a sentiment labeling project in Label Studio ▷ Prepare labeling interface
Configure Label Studio to use local LLMs through Ollama ▷ Connect Ollama backend
Choose model $m \in M$ ▷ Qwen 2.5 or Llama 3.2
for each comment $c_i \in C$ **do**
    Send c_i to model m via Ollama ▷ Inference
    Obtain predicted sentiment label $y_i \in \{\text{positive}, \text{negative}, \text{neutral}\}$ ▷ LLM prediction
    Display c_i and y_i in Label Studio ▷ Auto-annotation
    (Optional) Human verifies or corrects the label ▷ Review step
    Save (c_i, y_i) in Label Studio ▷ Store annotation
end for
Export all annotations from Label Studio ▷ Export labeled data
$D \leftarrow$ exported annotated dataset ▷ Final labeled corpus
return D ▷ Annotated sentiment dataset

Figure 4.3: Automatic sentiment annotation process using Label Studio and locally deployed large language models.

In addition, the test set was manually reviewed and corrected to ensure annotation quality and provide a reliable benchmark for the final evaluation of all sentiment classification models. Figure 4.4 presents the Streamlit application we made for manual review and correction of test-set annotations.

4.2.4 Data Preprocessing

Data preprocessing is a crucial step in natural language processing (NLP) tasks. Since social media comments contain spelling variations, Arabic expressions, emojis,

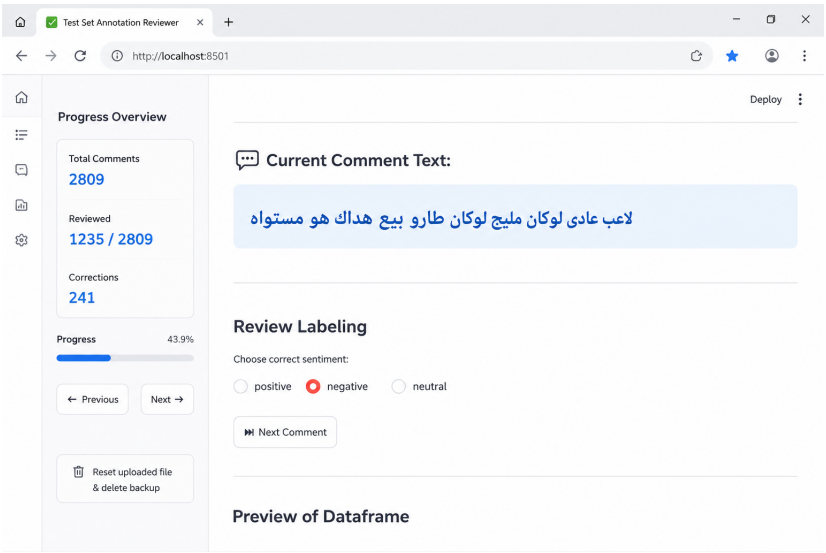


Figure 4.4: Streamlit application for manual review and correction of test-set annotations.

and informal writing styles, we applied several cleaning and normalization operations to reduce textual noise and improve data consistency.

The preprocessing steps we used are described as follows:

- **Missing value removal:** We removed comments with missing or empty text fields to ensure dataset quality and consistency.
- **Duplicate removal:** We removed duplicate comments to avoid redundancy and prevent potential bias during model training and evaluation.
- **Text cleaning:** We removed URLs, hashtags, mentions, extra spaces, and unnecessary noisy characters. We also converted the text to lowercase letters.
- **Emoji normalization:** We replaced emojis with text descriptions that express their meanings or emotions. For example, we converted emojis related to happiness, sadness, anger, or love into descriptive Arabic words.
- **Removing Arabic diacritics (Tashkil):** We removed Arabic diacritics to reduce variations in writing styles.
- **Arabic letter normalization:** We standardized different forms of certain Arabic letters to keep the spelling consistent, such as إ, أ, آ, and ا to ا, while ى was converted to ي.
- **Arabizi normalization:** We converted common Arabizi numerals into their Arabic letter equivalents, for example, 7 → ح, 2 → ء, 5 → خ, 9 → ق, 3 → ع, and 6 → ط.

- **Repeated character reduction:** We reduced sequences of repeated characters to minimize common spelling variations in social media comments, while keeping the original meaning of the word.
- **Tokenization:** After cleaning and normalization, we tokenized the comments for further processing and model training.

After preprocessing, we checked the sentiment distribution of the labeled corpus and applied a balancing procedure to reduce class imbalance between the sentiment categories. We then used the resulting dataset for representation learning and sentiment classification experiments.

4.3 Feature Extraction and Word Embedding Techniques

Machine learning and deep learning models cannot directly process textual data; therefore, after preprocessing, the comments were converted into numerical representations suitable for sentiment analysis of the Algerian dialect. We used various feature extraction and word embedding techniques to compare and evaluate their performance.

4.3.1 TF-IDF-Based Representations

We used Term Frequency–Inverse Document Frequency (TF-IDF) as a traditional text representation method to change text data into numerical feature vectors for sentiment classification. In this project, we tested three different TF-IDF-based representations.

First, we created a word-level TF-IDF representation to capture the importance of words and short phrases in the dataset. Second, we used a character-level TF-IDF representation based on character n-grams to better handle different spelling variations, dialect writing styles, and informal social media text. Finally, we built a hybrid TF-IDF representation by combining both word-level and character-level features into a single feature space.

We used these representations as input features for our traditional machine learning classifiers. They also served as strong baseline methods to compare against our word embedding and transformer-based approaches.

Representation	**Main Parameters**
Word TF-IDF	n-grams (1, 2), max features = 50,000
Character TF-IDF	character n-grams (3, 5), max features = 50,000
Hybrid TF-IDF	combination of word and character TF-IDF

Table 4.1: Parameters of the TF-IDF Feature Extraction Methods.

4.3.2 Word2Vec Embeddings

Word2Vec embeddings were trained on the unlabeled AlgeriaPulse corpus using the Skip-Gram architecture. After several experiments, the final model was trained with 150-dimensional vectors, a context window size of 5, a minimum word frequency threshold of 2 (min_count), and 10 training epochs. The resulting dense vector representations capture semantic relationships between words and were used as input features for sentiment classification models.

4.3.3 FastText Embeddings

FastText embeddings were trained on the unlabeled AlgeriaPulse corpus using the Skip-Gram architecture. The model was configured with 150-dimensional vectors, a context window size of 5, a minimum word frequency threshold of 2 (min_count), and 10 training epochs. Unlike Word2Vec, FastText represents words as a combination of character n-grams, enabling it to better handle spelling variations, Arabizi forms, dialectal expressions, and rare or unseen words frequently encountered in Algerian social media text.

4.3.4 Doc2Vec Embeddings

We used Doc2Vec embeddings to create representations for whole comments instead of single words. Unlike Word2Vec and FastText, which represent individual words, Doc2Vec generates a vector for the entire comment. This makes it very useful for sentiment classification at the comment level. In this work, we trained the model on the cleaned training, validation, and unlabeled comments. We configured the training using 150-dimensional vectors, a context window size of 5, a minimum word count threshold of 2, and 20 training epochs. We chose the Distributed Memory architecture, and we used the final document vectors with Logistic Regression for the sentiment classification phase.

4.3.5 Hybrid Representation

In this experiment, we trained FastText on the unlabeled, labeled, and test comments using the Skip-Gram architecture with 300-dimensional vectors, a context window size of 5, and a minimum word frequency threshold of 2. We obtained sentence-level FastText representations by averaging the word vectors of each comment. Then, we combined these dense vectors with the Character TF-IDF feature vectors into a single joined representation. We used the final hybrid vectors as input for a Linear Support Vector Machine classifier.

4.4 Model Training

After converting the comments into numerical representations, several classification models were used to perform sentiment analysis. These included machine learning algorithms, deep learning models, and pretrained language models.

4.4.1 Machine Learning Models

The sentiment classification dataset was divided into training and validation sets, while a separate manually annotated test set was reserved for the final evaluation. We explored several feature extraction and text representation techniques in this study, including TF-IDF-based representations, Word2Vec, FastText, Doc2Vec, and a fusion representation that combines Character TF-IDF and FastText features. We used these representations to capture lexical, morphological, and semantic information from the corpus.

We evaluated two traditional machine learning classifiers:

- **Logistic Regression (LR):** We tested Logistic Regression using Word TF-IDF, Word2Vec, FastText, and Doc2Vec representations. As a linear probabilistic classifier, it is widely used in sentiment analysis because it is simple, efficient, and performs well on high-dimensional text data.
- **Support Vector Machine (SVM):** We evaluated Support Vector Machine using Character TF-IDF, Hybrid TF-IDF, Word2Vec, FastText, and the fusion representation combining Character TF-IDF and FastText. SVM is very effective for text classification because it can find strong decision boundaries in high-dimensional feature spaces.

We selected these models because they are widely used and have proven effectiveness in sentiment classification tasks. Testing them with different feature representations allows for a complete comparison of sparse, dense, and hybrid text representations for Algerian dialect sentiment analysis.

4.4.2 Deep Learning Models

To test how well deep learning works for Algerian dialect sentiment analysis, we built a Bidirectional LSTM (BiLSTM) network. We used the Word2Vec and FastText embeddings that we trained earlier on the unlabeled AlgeriaPulse corpus to set up the embedding layer. Both of these embedding models were trained using 150-dimensional vectors.

We divided our labeled data into training and validation sets, and we kept a separate labeled test set for the final evaluation. We broke the comments into tokens, turned them into number sequences, and padded them so that all comments have the

same length. We also used one-hot encoding to represent our three sentiment labels, which are positive, negative, and neutral.

We tested both types of embeddings with our BiLSTM structure. The model begins with the embedding layer that uses the trained embedding vectors, followed by a Bidirectional LSTM layer with 128 hidden units. We put a dropout layer right after this to prevent overfitting. Next, we used a dense layer with 64 units and ReLU activation, and we finished with a 3-neuron Softmax layer for the final classification.

To train the model, we used the Adam optimizer and categorical cross-entropy loss. We ran the training for up to 15 epochs with a batch size of 64. We also included an early stopping feature based on validation loss, which stops the training process early and reloads the best weights so the model does not overfit.

Besides the BiLSTM structure, we also tested a Transformer Encoder model using our Word2Vec and FastText embeddings. This model also starts with an embedding layer set up with the trained embedding vectors. Next, it passes the data through a Transformer Encoder block, which uses Multi-Head Self-Attention, residual connections, layer normalization, and a feed-forward network. After that, we used a Global Average Pooling layer to group the sequence features together before sending them to a final fully connected layer and a Softmax layer for classification.

We trained the Transformer Encoder models using the Adam optimizer and categorical cross-entropy loss, keeping the exact same training, validation, and testing settings that we used for the BiLSTM experiments.

4.4.3 Transformer-Based Models

Beyond traditional machine learning and deep learning methods, we also explored transformer-based language models for Algerian dialect sentiment analysis. Unlike Word2Vec, FastText, and Doc2Vec, these models can understand context, which means that the representation of a word changes depending on the words around it.

We fine-tuned four pretrained transformer models, which are DziriBERT, XLM-RoBERTa (XLM-R), MARBERT, and AraBERT, on our training dataset. We used the validation set to check the performance during training, and we kept a separate, manually annotated test set for the final evaluation.

To prepare the data, we tokenized the comments using the specific tokenizer that belongs to each model. These tokenized sequences were then used as input for the transformer model, followed by a classification layer to predict one of the three sentiment classes: positive, negative, or neutral.

For the fine-tuning process, we used the AdamW optimizer and cross-entropy loss. We trained the models for up to 3 epochs, and we selected the batch size based on our available computing resources. The best model checkpoint was automatically chosen based on the validation results and saved for the final evaluation.

The goal of this experiment was to compare domain-specific embeddings, deep learning structures, and large pretrained language models. This helped us find the most effective approach for analyzing sentiment in the Algerian dialect.

4.5 Evaluation Metrics

Several standard evaluation metrics were used to measure the performance of the sentiment classification models. Because the task involves three sentiment categories (positive, negative, and neutral), accuracy alone may not give a complete picture of model effectiveness, so Precision, Recall, and Macro F1-score were also included [49].

Accuracy reflects the percentage of comments correctly labeled out of the total dataset [49].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Precision indicates how many of the samples predicted as belonging to a particular class were actually correct [49].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

Recall shows how many of the true instances of a class were successfully identified by the model [49].

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

Macro F1-score is calculated as the harmonic mean of Precision and Recall, computed separately for each class and then averaged across all classes. This approach assigns equal weight to every sentiment category and is particularly valuable when class distributions are skewed [49].

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

$$\text{Macro-F1} = \frac{1}{C} \sum_{i=1}^{C} F_{1i} \tag{4.5}$$

Confusion matrices were further analyzed to provide deeper insight into misclassification patterns and how predictions were distributed across the sentiment classes [49]. Final model comparisons relied on results obtained from the manually annotated test set.

where (TP), (TN), (FP), and (FN) denote the numbers of true positives, true negatives, false positives, and false negatives, respectively, and (C) represents the total number of sentiment classes.

4.6 Majority Voting

To improve the accuracy of our sentiment analysis framework, we implemented an ensemble learning approach. Instead of relying on a single model, we combine predictions from multiple diverse models using Majority Voting. This technique merges the strengths of different architectures to minimize individual errors. Within the broad field of consensus aggregation, ensemble methods are primarily divided into two practical voting types: Hard Voting and Soft Voting [50].

4.6.1 Hard Voting

Hard voting is a simple majority rule. Every independent model gets one equal vote. Each model outputs a single sentiment label (Positive, Neutral, or Negative), and the system selects the label that receives the highest number of votes [50].

Formally, let N be the total number of models, and y_i be the class label predicted by the i-th model. The final ensemble prediction $\hat{y}_{\text{hard}}$ is defined as:

$$\hat{y}_{\text{hard}} = \text{mode}(y_1, y_2, \ldots, y_N) \tag{4.6}$$

4.6.2 Soft Voting

Soft voting takes into account how sure each model is by using confidence scores instead of simple votes. It predicts the final class label by averaging the probability scores assigned to each sentiment category by every individual model [50].

Let $P_i(y = c)$ denote the confidence score assigned to a specific class c by the i-th model. The final soft voting prediction $\hat{y}_{\text{soft}}$ is calculated by choosing the class with the highest average probability:

$$\hat{y}_{\text{soft}} = \arg\max_c \frac{1}{N} \sum_{i=1}^{N} P_i(y = c) \tag{4.7}$$

4.6.3 Voting Strategy Adopted in This Work

Both hard voting and soft voting are widely used ensemble learning strategies. The main difference between them is that hard voting determines the final prediction based on the class labels predicted by each model, whereas soft voting combines the confidence scores or class probabilities produced by the individual models. Although soft voting can exploit probability information, it requires all participating models to provide reliable and well-calibrated confidence scores.

In this work, we adopted the Hard Majority Voting strategy because it provides a simple, transparent, and model-independent mechanism for combining the predictions of heterogeneous sentiment classification models. Our ensemble consists of FastText + Transformer, Word2Vec + Transformer, AraBERT, MARBERT, XLM-RoBERTa,

DziriBERT, and the Gemini API. Each model contributes one vote, and the sentiment class receiving the highest number of votes is selected as the final prediction.

Combining these heterogeneous models allows the ensemble to benefit from their complementary strengths. The embedding-based transformer models capture semantic representations learned from the AlgeriaPulse corpus, while the pretrained transformer models contribute contextual knowledge acquired from large-scale Arabic corpora. In addition, Gemini provides an independent large language model capable of reasoning over the input text. By aggregating the predictions of these models, the Hard Majority Voting ensemble reduces the impact of individual model errors and improves the robustness and reliability of the final sentiment prediction. Figure 4.5 illustrates the Hard Majority Voting mechanism adopted in this work.

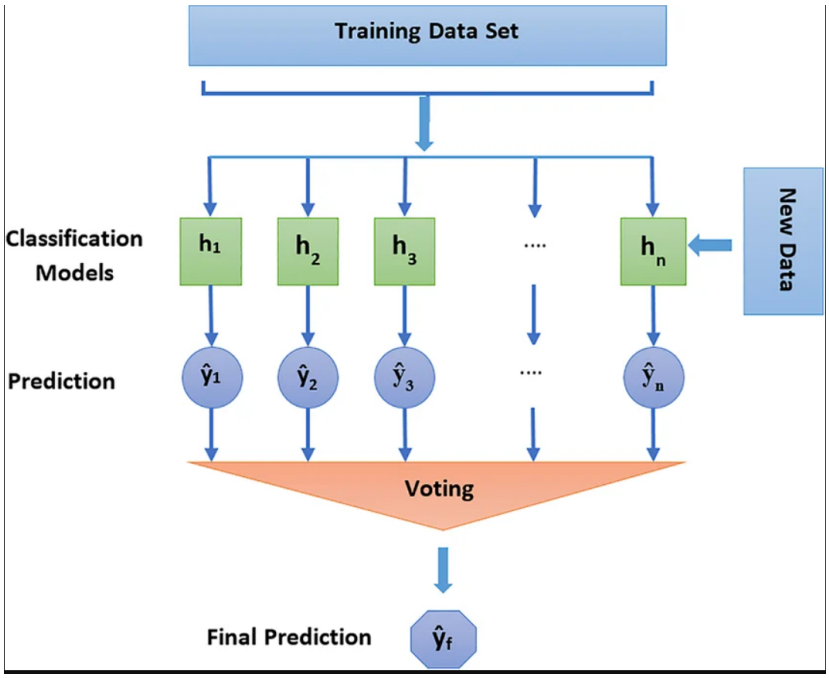


Figure 4.5: Illustration of the Hard Voting Ensemble Method.

Conclusion

In this chapter, we presented the methodology used for sentiment analysis in the Algerian dialect. We described how we built the AlgeriaPulse corpus through data collection, annotation, dataset integration, balancing, and preprocessing. We also explained how we developed the labeled and unlabeled resources used throughout this study. We investigated several text representation techniques, including TF-IDF-based representations, as well as Word2Vec, FastText, and Doc2Vec embeddings trained on our unlabeled corpus. These representations were then used as input features for our traditional machine learning and deep learning models. Additionally, we explored pretrained transformer-based language models to evaluate the effectiveness of contextual

representations for sentiment classification. This chapter also detailed the classification models considered in this work, including machine learning algorithms, deep learning architectures, and pretrained transformer-based approaches. Finally, we introduced the proposed Hard Majority Voting ensemble, which combines the predictions of the best-performing models, including the Gemini API, to improve the robustness and reliability of the final sentiment classification. We also presented the evaluation metrics used to assess and compare the performance of all the proposed approaches. The experimental results and comparative analysis of these methods are presented in the next chapter.

Chapter 5

Implementation and Results

Introduction

This chapter presents the implementation and experimental results of the sentiment analysis models developed in this study. Following the preparation of the AlgeriaPulse dataset and the methodology described in the previous chapter, we present the experimental setup, the implementation of the different sentiment classification models, and the obtained results. The evaluated models include traditional machine learning algorithms, the BiLSTM deep learning model, and transformer-based pretrained language models. We also assess the effectiveness of the majority voting ensemble approach by combining the predictions of multiple classifiers. Finally, we compare the performance of all models using the evaluation metrics Accuracy, Precision, Recall, and F1-Score and discuss their effectiveness for sentiment analysis of the Algerian dialect.

5.1 Experimental Environment

5.1.1 Hardware Resources

All experiments were conducted using the cloud-based *Kaggle Notebooks* environment. The platform operated on Linux (Kernel 6.6.122+) with Python 3.12.13. The available hardware consisted of a 4-core `x86_64` CPU, 31.35 GB of RAM, and two NVIDIA Tesla T4 GPUs, each equipped with 15 GB of GPU memory. These GPUs were utilized to accelerate the training of the deep learning and transformer-based models, significantly reducing the overall training time and enabling efficient experimentation on large-scale datasets.

5.1.2 Programming Language

Python was used as the primary programming language throughout this research due to its simplicity, flexibility, readability, and extensive ecosystem of scientific computing and machine learning libraries. It has become one of the most widely adopted programming languages for data science, machine learning, and Natural Language Processing (NLP), providing efficient tools for data preprocessing, feature extraction, model development, and evaluation.

5.1.3 Libraries and Frameworks

Several Python libraries and frameworks were employed throughout this work to support the different stages of implementation, including data preprocessing, corpus

analysis, feature extraction, model training, evaluation, and visualization. The main libraries used are summarized below.

- **Pandas**: Used for loading, cleaning, filtering, and organizing the datasets.
- **NumPy**: Used for numerical computations, array manipulation, and matrix operations.
- **Re (Regular Expressions)**: Applied during the preprocessing stage to normalize textual data by removing unwanted patterns and formatting inconsistencies.
- **Matplotlib & Seaborn**: Used to generate visualizations, including sentiment distribution charts, corpus analysis figures, model comparison plots, and confusion matrices.
- **Scikit-learn (Feature Extraction and Classification)**: Used to generate TF-IDF, Character TF-IDF, and Hybrid TF-IDF representations, as well as to implement the Logistic Regression and Support Vector Machine (SVM) classifiers.
- **Scikit-learn (Evaluation)**: Used for dataset splitting and for calculating the evaluation metrics, including Accuracy, Precision, Recall, and F1-Score.
- **Gensim**: Used to train the Word2Vec, FastText, and Doc2Vec embedding models on the unlabeled AlgeriaPulse corpus.
- **SciPy**: Used for sparse matrix operations and feature fusion to construct the hybrid representation combining Character TF-IDF features with FastText sentence embeddings.
- **TensorFlow & Keras**: Used to implement and train the Bidirectional Long Short-Term Memory (BiLSTM) models by integrating the pretrained Word2Vec and FastText embeddings.
- **Hugging Face Transformers**: Used for fine-tuning and evaluating the pretrained transformer models, including AraBERT, MARBERT, XLM-RoBERTa, and DziriBERT.
- **Hugging Face Datasets**: Used for efficient dataset preparation, tokenization, and data management during transformer model training.
- **PyTorch**: Used as the underlying deep learning framework for training and inference with all transformer-based models.

Together, these libraries provided the necessary infrastructure to implement, train, evaluate, and compare traditional machine learning models, deep learning architectures, transformer-based models, and the majority voting ensemble approach for sentiment analysis of the Algerian dialect.

5.2 Corpus Analysis

5.2.1 Corpus Statistics

5.2.1.1 Unlabeled Corpus Statistics

The unlabeled AlgeriaPulse corpus contains around 1.47 million comments collected from Algerian social media platforms. After preprocessing, the dataset consists of nearly 17 million words and over 695,000 unique tokens.

This high number of unique words shows the linguistic diversity of Algerian online text, which includes dialectal vocabulary, Arabizi expressions, spelling variations, named entities, and topic-specific terms. This lexical richness makes the corpus a great resource for training word representation models, as it captures semantic relationships specific to the Algerian dialect.

Table 5.1: Main Statistics of the AlgeriaPulse Unlabeled Corpus

Statistic	**Value**
Number of comments	1,476,896
Total words	16,989,921
Unique words	695,263

5.2.1.2 Labeled Corpus Statistics

The labeled sentiment corpus went through several processing stages to create the final dataset for this study. Initially, the corpus had 133,422 comments. To improve the quality of the data and keep it consistent, we removed all duplicate entries and missing values, which reduced the size to 84,716 comments. Next, we checked the sentiment distribution and applied a balancing procedure to fix the class imbalance problem. This step left us with a final labeled corpus of 71,211 comments.

Table 5.2 shows how the dataset changed through these different processing stages.

Table 5.2: Evolution of the Labeled Corpus

Stage	**Number of comments**
Initial labeled corpus	133,422
After duplicate and missing value removal	84,716
Final balanced corpus	71,211

Looking at the final results, the corpus consists of 25,397 positive comments, 25,098 negative comments, and 20,716 neutral comments, as shown in Figure 5.1.

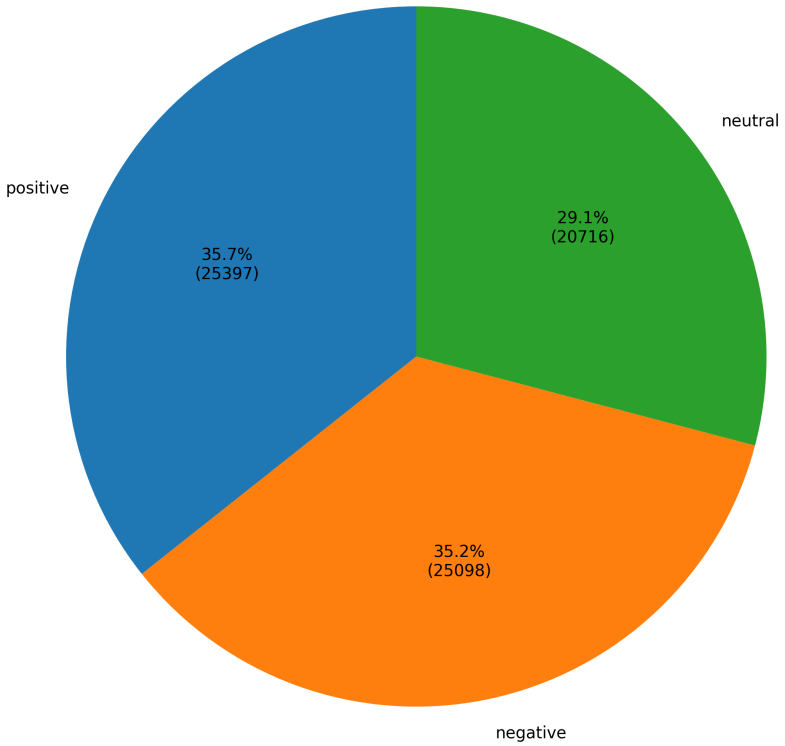


Figure 5.1: Final Sentiment Distribution

5.2.1.3 Train, Validation, and Test Set

We split our final labeled dataset of 71,211 comments into training and validation sets using an 80%-20% stratified split. The training set was used to train our models, while the validation set helped us choose the best model and monitor performance during training. We also kept an independent test set of 4682 manually reviewed comments completely separate, using it only for the final evaluation of all our sentiment classification models.

Table 5.3 shows the exact number of positive, negative, and neutral comments in the training, validation, and test sets.

Table 5.3: Class Distribution of the Training, Validation, and Test Sets

Dataset Split	**Positive**	**Negative**	**Neutral**	**Total**
Training	20,317	20,078	16,573	56,968
Validation	5,080	5,020	4,143	14,243
Test	1360	1817	1505	4682

5.2.2 Bigram Analysis

While individual words provide helpful information about sentiment, they often fail to capture how terms relate to each other when they appear together. To better understand the linguistic patterns used by Algerian social media users, we conducted a bigram analysis. Bigrams consist of two consecutive words, allowing us to identify common expressions that frequently appear within each sentiment category.

We ran this analysis separately on positive and negative comments after completing the dataset preprocessing.

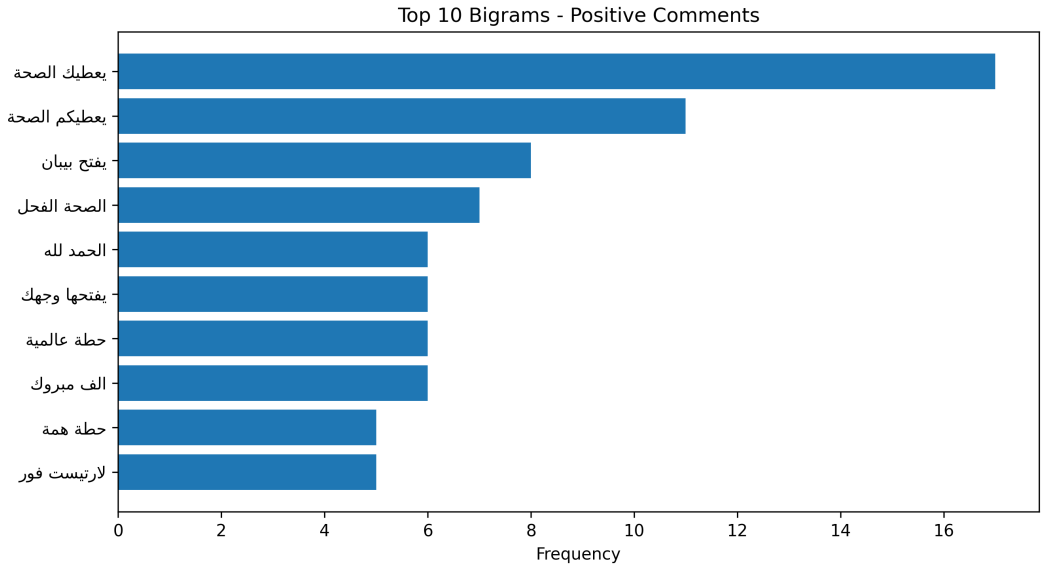


Figure 5.2: Top 10 Positive Bigrams

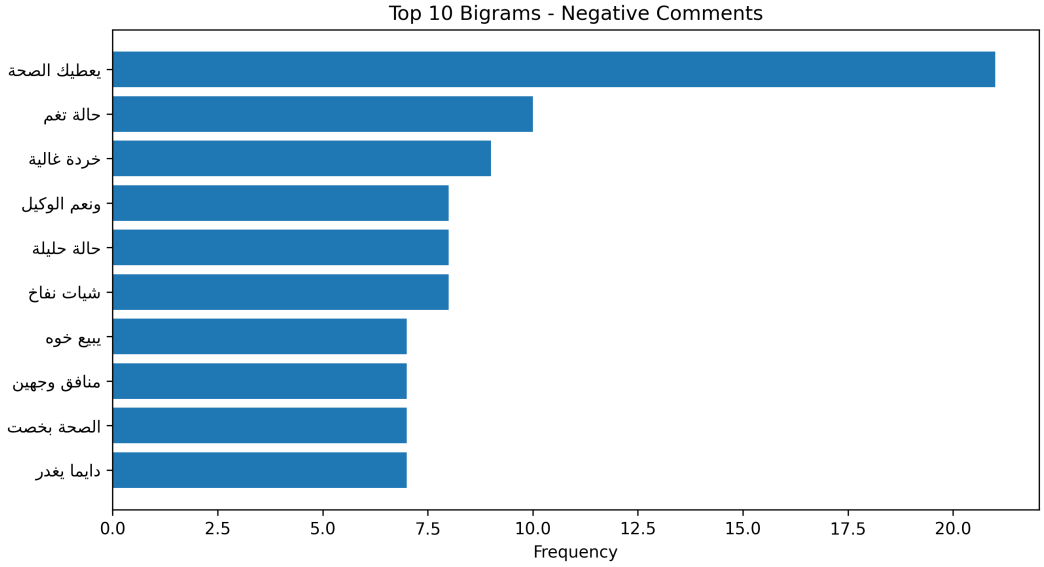


Figure 5.3: Top 10 Negative Bigrams

As shown in Figures 5.2 and 5.3, the positive bigrams are dominated by ex-

pressions of appreciation and encouragement. Phrases like يعطيك الصحة and يعطيكم الصحة appear at the top, which are commonly used in the Algerian dialect to thank someone. Other frequent terms include blessings and congratulations like الحمد لله and ألف مبروك, alongside casual compliments like حطة عالمية and لارتيست فور.

In contrast, the negative bigrams mainly reflect criticism and frustration. Expressions like خردة غالية (expensive junk) and حالة تغم show clear dissatisfaction, while terms like شيات نفاخ, يبيع خوه, and منافق وجهين represent direct personal insults. Additionally, religious phrases like ونعم الوكيل are used here to convey a deep sense of grievance.

An interesting observation is that the expression يعطيك الصحة appears as the top bigram in both categories. While it is traditionally a positive phrase, its high frequency in the negative subset shows it is heavily used for sarcasm. This overlap proves that the true sentiment depends entirely on context, highlighting the challenge of analyzing sarcasm in informal Algerian social media data.

5.2.3 Trigram Analysis

While bigrams provide helpful information about common word combinations, they are still too limited to capture complete expressions. Therefore, we performed a trigram analysis to identify the most frequent three-word sequences in both positive and negative comments. Trigrams preserve a larger portion of the linguistic context, offering a more detailed representation of how sentiments are expressed in Algerian social media communication.

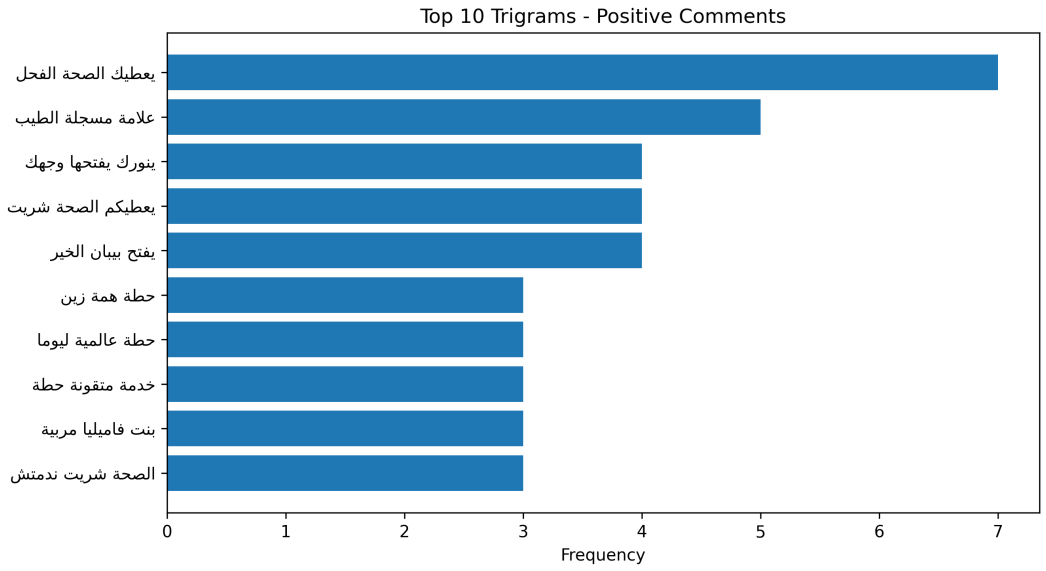


Figure 5.4: Top 10 Positive Trigrams

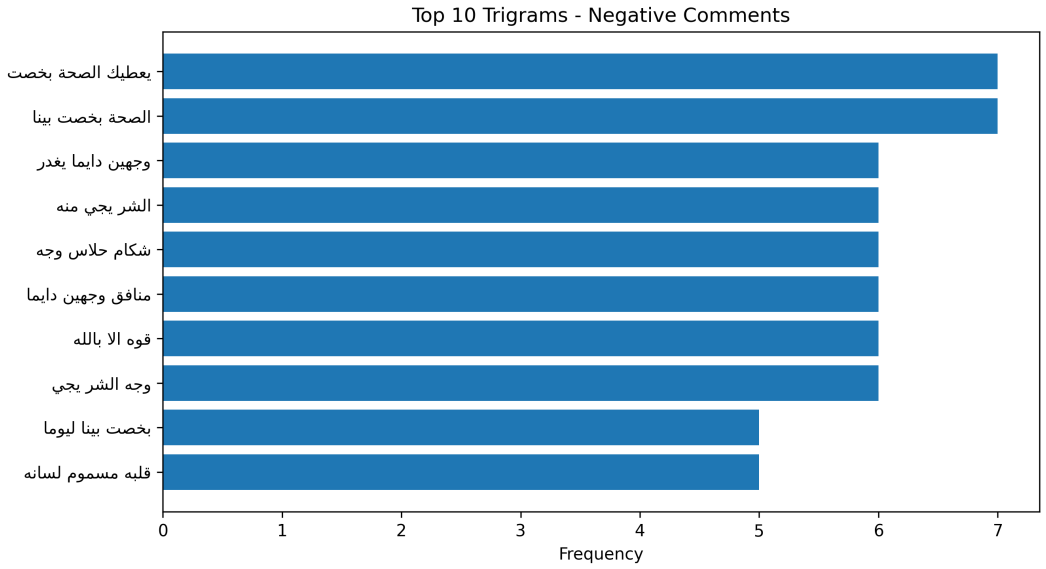


Figure 5.5: Top 10 Negative Trigrams

As shown in Figures 5.4 and 5.5, the positive trigrams are dominated by cultural expressions of praise and satisfaction. The top phrase is يعطيك الصحة الفحل, showing deep appreciation. Other frequent sequences include blessings like ينورك يفتحها وجهك and يفتح بيبان الخير, alongside satisfaction remarks like الصحة شربت ندمتش (I bought it and don't regret it) and compliments like بنت فاميليا مربية and حطة عالمية ليوما.

Conversely, the negative trigrams consist of clear mockery, frustration, and insults. The top sequences are يعطيك الصحة بخصت and الصحة بخصت بينا (you humiliated us), which appear alongside بخصت بينا ليوما. Personal attacks on integrity are also highly visible, including وجهين دايما يغدر, منافق وجهين دايما, and شكام حلاس وجه.

Crucially, this trigram analysis confirms why the bigram يعطيك الصحة appeared in both classes. While the bigram alone is ambiguous, the trigram expansion reveals the exact context. In the positive class, it expands into genuine compliments like يعطيك الصحة الفحل, whereas in the negative class, it expands into sarcastic remarks like يعطيك الصحة بخصت. This proves that longer sequences are essential to capture the contextual modifiers that turn a positive phrase into sarcasm.

5.2.4 Word Cloud Visualization

Word clouds were generated to provide a visual representation of the most frequent terms in each sentiment category, where the size of each word corresponds to its frequency within the corresponding subset of comments. This visualization offers an intuitive overview of the vocabulary associated with each sentiment and complements the quantitative analyses presented in the previous sections.

Figure 5.6 illustrates the word cloud generated from positive comments, while Figure 5.7 presents the word cloud corresponding to negative comments.



Figure 5.6: Positive Word Cloud

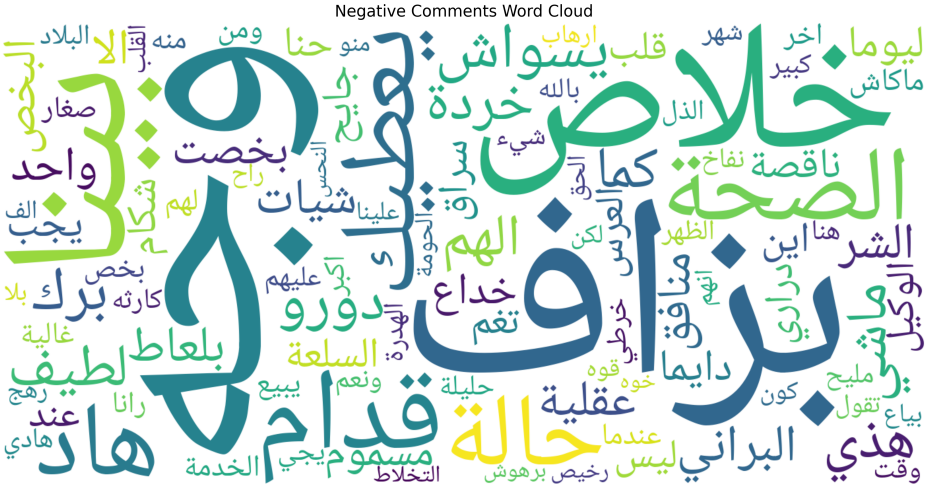


Figure 5.7: Negative Word Cloud

The positive word cloud highlights terms related to appreciation, encouragement, gratitude, and religious blessings, reflecting the supportive and optimistic nature of positive comments. In contrast, the negative word cloud is dominated by words expressing criticism, dissatisfaction, and disagreement. These visualizations further confirm the lexical differences between sentiment categories observed in the bigram and trigram analyses.

5.3 Representation Statistics

This section shows the statistical features of the text representations we made from the AlgeriaPulse corpus. These statistics give an overview of the feature spaces and vocabularies we got from the different representation methods used in our sentiment classification experiments.

As shown in Table 5.4, the TF-IDF representations created high-dimensional sparse feature spaces, while Word2Vec, FastText, and Doc2Vec gave us dense vector representations. Our Fusion representation combines the 50,000 Character TF-IDF features with 300-dimensional FastText sentence embeddings. This setup allows us to capture both lexical and semantic information at the same time.

Table 5.4: Main Statistics of the Generated Text Representations

Representation	**Vocabulary / Features**
TF-IDF	50,000 features
Character TF-IDF	50,000 features
Hybrid TF-IDF	100,000 features
Word2Vec	182,376 words
FastText	182,376 words
Doc2Vec	182,376 words
Fusion Representation	50,300 features

5.4 Experimental Results

5.4.1 Machine Learning Models

Figure 5.8 shows the performance scores we got from the machine learning models tested on the AlgeriaPulse corpus. Our experiments cover traditional TF-IDF representations mixed with Logistic Regression, word embedding methods like Word2Vec, FastText, and Doc2Vec, plus character-level and hybrid features paired with Support Vector Machines (SVM). The results reveal clear differences between the text representation methods and classifiers, which proves how important it is to pick the right features when analyzing Algerian dialect sentiment.

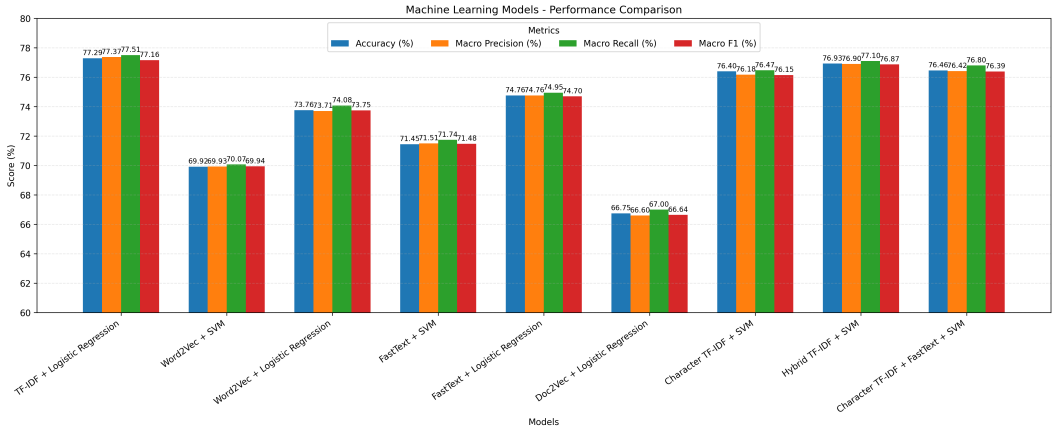


Figure 5.8: Bar chart accuracy comparison of machine learning models

As shown in Figure 5.8, the TF-IDF representation combined with Logistic Regression achieved the highest accuracy among the machine learning models, with

77.29% accuracy and 77.16% Macro F1. It was followed by Hybrid TF-IDF + SVM with 76.93% accuracy, Character TF-IDF + FastText + SVM with 76.46%, and Character TF-IDF + SVM with 76.40%. The close results between these models show that TF-IDF and character-level representations are very useful for the Algerian dialect, especially because our comments contain spelling variations, informal writing, and mixed language.

FastText + Logistic Regression also gave a good result, reaching 74.76% accuracy, while Word2Vec + Logistic Regression achieved 73.76%. With SVM, FastText reached 71.45%, while Word2Vec reached 69.92%. This shows that FastText was slightly better than Word2Vec in these experiments, probably because it uses subword information, which helps with dialectal words and spelling differences.

On the other hand, Doc2Vec + Logistic Regression gave the lowest result, with 66.75% accuracy and 66.64% Macro F1. This suggests that document-level embeddings were less suitable for our short social media comments. Overall, we can say that the machine learning models gave strong baseline results, especially when using TF-IDF and character-based features. These results are important because they show that simple and interpretable models can still perform well before moving to deep learning approaches.

5.4.2 Deep Learning Models Performance

Figure 5.9 shows the performance scores we got from the deep learning models tested on the AlgeriaPulse sentiment classification task. For these experiments, we paired two types of embedding methods—Word2Vec and FastText—with two different neural network architectures: Bidirectional Long Short-Term Memory (BiLSTM) and Transformer networks.

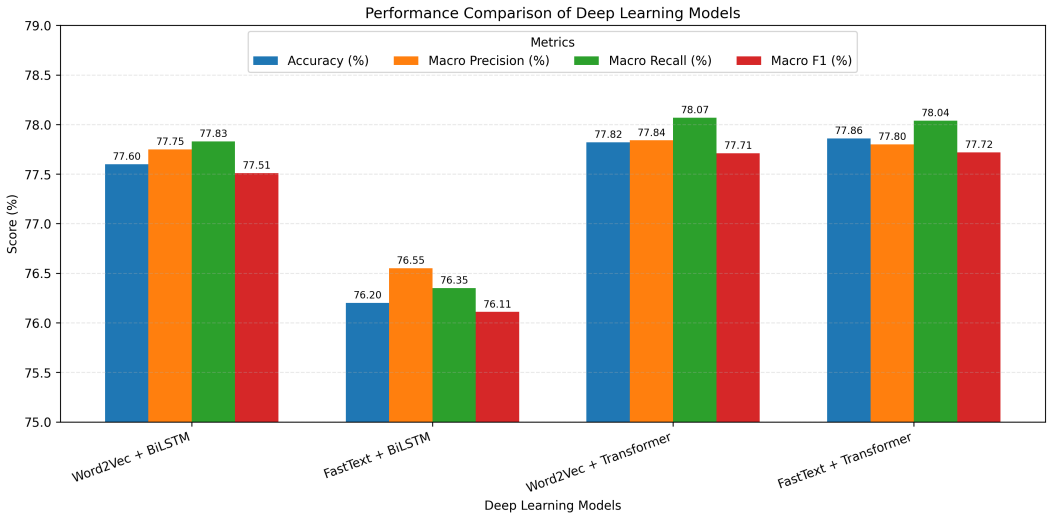


Figure 5.9: Bar chart accuracy comparison of deep learning models

The FastText + Transformer model achieved the best result, with an accuracy

of 77.86% and a Macro F1-score of 77.72%. The Word2Vec + Transformer model gave almost the same performance, reaching 77.82% accuracy and 77.71% Macro F1. This shows that both Transformer-based models performed very similarly, and that the Transformer architecture was able to capture useful contextual information from Algerian dialect comments.

For the BiLSTM models, Word2Vec + BiLSTM achieved 77.61% accuracy and 77.51% Macro F1, while FastText + BiLSTM reached 76.22% accuracy and 76.11% Macro F1. These results are still strong, but they are slightly lower than the Transformer models. This suggests that the self-attention mechanism used in Transformers gives a small advantage compared with the recurrent structure of BiLSTM.

We can also notice that the difference between Word2Vec and FastText is not very large in the deep learning experiments. In the Transformer models, the results are almost identical, while in the BiLSTM models, Word2Vec performed slightly better than FastText. This means that the neural architecture itself plays an important role in learning the sentiment patterns, not only the embedding type.

Overall, our deep learning models showed stable and competitive performance, with all models scoring above 76%. The best configuration was FastText + Transformer, but the difference with Word2Vec + Transformer was very small. This confirms that combining word embeddings with deep neural models is effective for sentiment classification in Algerian social media text.

5.4.3 Pretrained Transformer Models Performance

Figure 5.10 compares the performance of the pretrained transformer models tested on the AlgeriaPulse dataset. The results show a clear improvement over both our traditional machine learning and deep learning approaches, which highlights how effective contextual language representations are for Algerian dialect sentiment analysis.

Among the pretrained transformer models, DziriBERT achieved the highest performance, reaching an accuracy of 82.25% with a Macro F1-score of 82.12%. This result demonstrates the advantage of using a language model specifically developed for the Algerian dialect, as it is better adapted to the vocabulary, expressions, and writing styles commonly found in Algerian social media comments.

MARBERT obtained the second-best performance with an accuracy of 81.56% and a Macro F1-score of 81.25%, followed by AraBERT, which achieved 80.49% accuracy and 80.30% Macro F1. Both models produced strong and consistent results, confirming that Arabic pretrained language models are highly effective for sentiment classification. However, they still performed below DziriBERT, suggesting that models trained on general Arabic data may not fully capture the linguistic characteristics of the Algerian dialect.

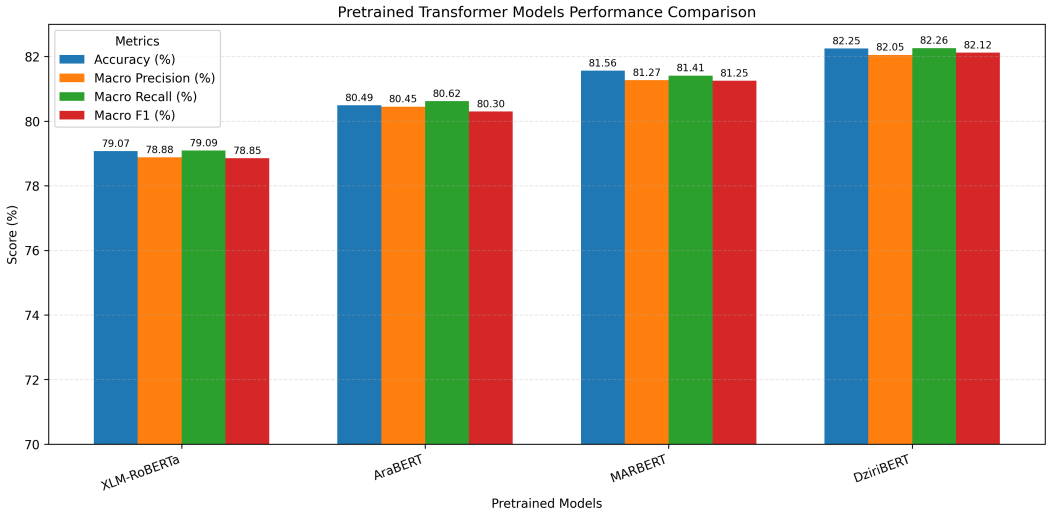


Figure 5.10: Bar chart accuracy comparison of pretrained transformer models

On the other hand, XLM-RoBERTa obtained the lowest performance among the pretrained models, achieving an accuracy of 79.07% and a Macro F1-score of 78.85%. Although it benefits from multilingual pretraining across many languages, its results indicate that language-specific pretrained models are generally more suitable for Algerian dialect sentiment analysis than large multilingual models.

Overall, the pretrained transformer models consistently outperformed all of our traditional machine learning and deep learning approaches. These findings highlight the importance of contextual representations and domain-specific pretraining, with DziriBERT emerging as the best-performing *individual* model for Algerian dialect sentiment classification.

5.4.4 Performance Analysis of the Best Individual Model

Among all the individual sentiment classification models evaluated in this study, DziriBERT achieved the best performance. To further analyze its classification behavior, a class-level evaluation was conducted for the three sentiment categories (negative, neutral, and positive), as presented in Table 5.5.

Table 5.5: Class-wise Performance of DziriBERT

Sentiment	**Precision**	**Recall**	**F1-score**	**Support**
Negative	84.30%	84.81%	84.55%	1817
Neutral	78.13%	73.82%	75.91%	1505
Positive	83.73%	88.16%	85.89%	1360

The results show that DziriBERT performs strongly on both negative and positive comments, achieving F1-scores of 84.55% and 85.89%, respectively. The neutral class obtained a lower F1-score of 75.91%, showing that it is more difficult to classify. This is expected because neutral comments often contain weaker sentiment signals and may

be confused with mildly positive or negative comments.

Figure 5.11 presents the confusion matrix of the DziriBERT model on the test set. The model correctly classified 1,541 out of 1,817 negative comments, 1,111 out of 1,505 neutral comments, and 1,199 out of 1,360 positive comments. Most classification errors occurred between the neutral class and the other two sentiment categories. Specifically, 240 neutral comments were misclassified as negative, while 154 were predicted as positive. Similarly, 197 negative comments and 114 positive comments were incorrectly classified as neutral. In contrast, direct confusion between the negative and positive classes was relatively limited, with only 79 negative comments predicted as positive and 47 positive comments predicted as negative. Overall, these results demonstrate that DziriBERT is highly effective at recognizing clear sentiment polarity, although distinguishing neutral comments remains comparatively more challenging.

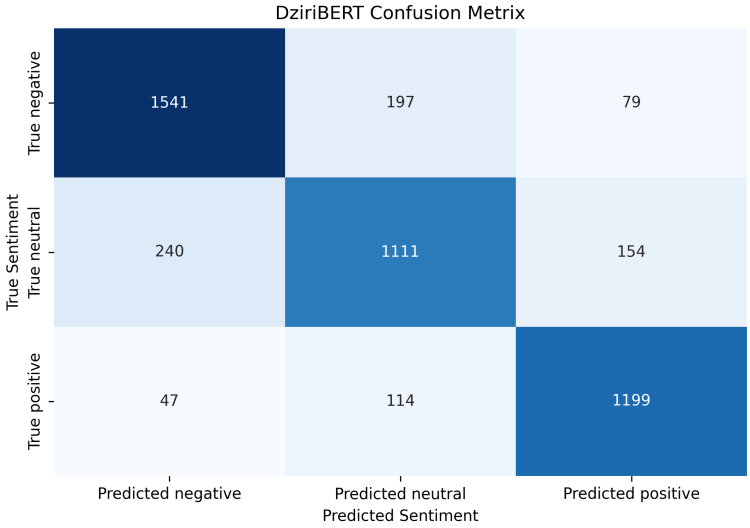


Figure 5.11: Confusion Matrix of the DziriBERT Model on the Test Set

5.5 Error Analysis

To better understand where DziriBERT fails, we examined the comments that were misclassified on the test set. Analyzing these prediction errors helps identify the main challenges faced by the model and provides insights into the linguistic characteristics that make sentiment classification more difficult. Table 5.6 presents representative examples of misclassified comments, including their true sentiment labels and the labels predicted by DziriBERT.

The error examples presented in Table 5.6 illustrate several challenges associated with Algerian dialect sentiment analysis. First, some misclassifications are caused by religious expressions. For example, comments containing prayers such as الله يشافيه وعافيه آمين يا رب العالمين were predicted as positive even though they were labeled as neutral. Similarly, the model was misled by comments containing positive lexical cues

Table 5.6: Representative Misclassified Comments by DziriBERT

Comment	**True Label**	**Predicted Label**	**Main Challenge**
الله يشافيه وعافيه آمين يا رب العالمين	Neutral	Positive	Religious prayer expression
الناس تتكلم عن الصناعات والتكنولوجية ونحن العرب نتعارك عن الكسكوس	Negative	Neutral	Implicit criticism without explicit negative words
شكون ليسمعها في 2020	Neutral	Positive	Positive emojis in a short context-dependent comment
يعطيك الصحة وكلتهم وعرفتهم على تقاليدنا بصح دايمن تغيضني كي دزيريين ما يعرفوش يشرحوا للاجانب أطباقنا	Positive	Negative	Mixed sentiment within the same comment
الحمد لله كاين بديل خير من وجهك الحمد لله راك في الحبس الله لا تربحك	Negative	Positive	Religious words and the positive word *"khir"* (better) masking negative sentiment

such as الحمد لله and the word خير (better), despite the overall sentiment being negative.

Another common challenge is implicit sentiment. Some comments express criticism without using explicit negative words. For example, the comment الناس تتكلم عن الصناعات والتكنولوجية ونحن العرب نتعارك عن الكسكوس conveys dissatisfaction through comparison rather than direct negative vocabulary, making it difficult for the model to correctly identify the sentiment.

The model also struggles with comments that contain mixed sentiment. In the example يعطيك الصحة وكلتهم وعرفتهم على تقاليدنا بصح دايمن تغيضني كي دزيريين ما يعرفوش يشرحوا للاجانب أطباقنا, the comment begins with appreciation but later expresses criticism. Such comments contain both positive and negative cues, making sentiment classification more challenging.

Finally, short comments and social media elements such as emojis can also lead to prediction errors. Comments like شكون ليسمعها في 2020 provide very limited textual context, while the presence of positive emojis may bias the model toward predicting a positive sentiment even when the overall meaning is neutral.

Overall, these errors show that sentiment analysis for the Algerian dialect requires more than identifying positive or negative words. Correctly classifying sentiment often depends on understanding context, implicit meanings, cultural and religious expressions, mixed opinions, and the informal nature of social media language.

5.6 Performance Evaluation of the Hard Majority Voting Ensemble

Following the overall comparison of all evaluated approaches, this section provides a detailed analysis of the proposed Hard Majority Voting ensemble. The evaluation focuses on its class-wise performance and confusion matrix to better understand how effectively the ensemble classifies the three sentiment categories.

Table 5.7: Class-wise Performance of the Hard Majority Voting Ensemble

Sentiment	**Precision**	**Recall**	**F1-score**	**Support**
Negative	87.93%	85.95%	86.93%	1817
Neutral	87.67%	86.49%	87.08%	1505
Positive	82.94%	86.90%	84.88%	1360

Table 5.7 presents the class-wise performance of the Hard Majority Voting ensemble. The ensemble achieved its highest F1-score for the neutral class (87.08%), followed closely by the negative class (86.93%). The positive class obtained an F1-score of 84.88%, making it the most challenging sentiment category to classify.

For the negative class, the ensemble achieved a Precision of 87.93% and a Recall of 85.95%, indicating that most negative comments were correctly identified while only a limited number were confused with the other classes. The neutral class obtained the highest Precision (87.67%) and Recall (86.49%), demonstrating that the ensemble effectively distinguished neutral comments from both positive and negative opinions. Although the positive class achieved the lowest Precision (82.94%), its Recall reached 86.90%, showing that most positive comments were successfully detected despite a small number of false positive predictions.

Figure 5.12 illustrates the confusion matrix of the Hard Majority Voting ensemble.

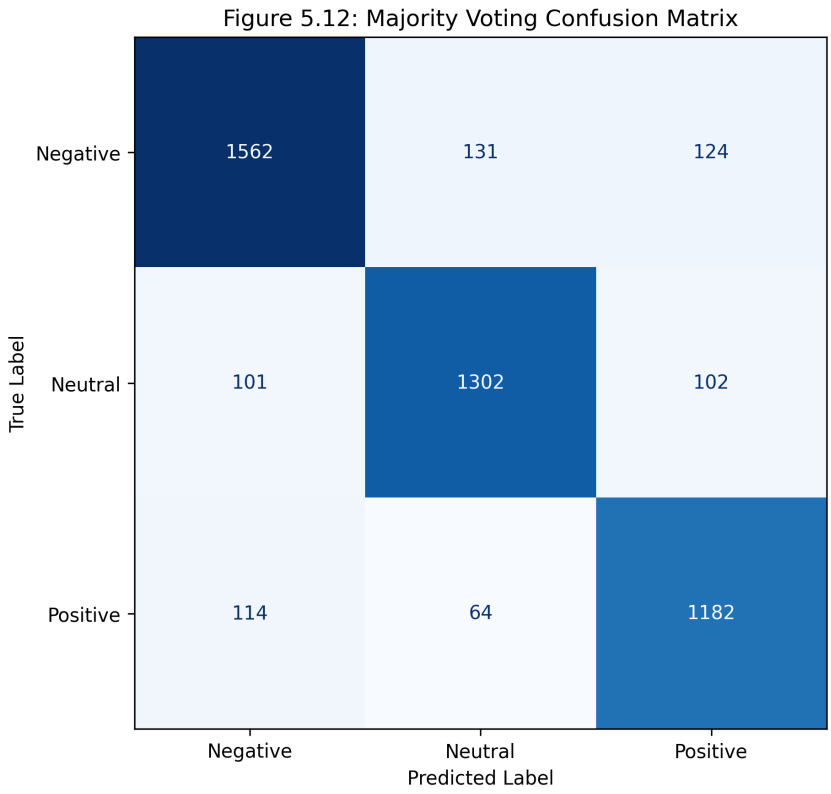


Figure 5.12: Confusion Matrix of the Hard Majority Voting Ensemble

As shown in Figure 5.12, the ensemble correctly classified 1,562 out of 1,817 negative comments, 1,302 out of 1,505 neutral comments, and 1,182 out of 1,360 positive comments. Most classification errors occurred between the neutral and positive classes. Specifically, 104 neutral comments were predicted as positive and 99 positive comments were predicted as neutral. Similarly, 146 negative comments were misclassified as neutral, while only 109 negative comments were directly classified as positive. Direct confusion between the negative and positive classes remained relatively limited, indicating that the ensemble effectively distinguished comments with opposite sentiment polarities.

Overall, the class-wise evaluation and confusion matrix demonstrate that the proposed Hard Majority Voting ensemble provides balanced and reliable sentiment classification across all three sentiment categories. These findings are consistent with the overall performance presented in Table 5.8, where the ensemble achieved the highest Accuracy (86.42%), Macro Precision (86.27%), Macro Recall (86.46%), and Macro F1-score (86.36%) among all evaluated approaches.

5.6.1 Overall Model Comparison

This summary table presents the performance of all the sentiment classification approaches evaluated in this study, including traditional machine learning, deep learning, pretrained transformer models, and the proposed Hard Majority Voting en-

semble. The results show a clear improvement in classification performance as the models become more advanced and better at capturing the linguistic characteristics of the Algerian dialect.

Table 5.8 presents the overall performance comparison of all evaluated approaches using Accuracy, Macro Precision, Macro Recall, and Macro F1-score. It provides a comprehensive overview of the effectiveness of each approach and facilitates a direct comparison between traditional machine learning, deep learning, pretrained transformer models, and the proposed Hard Majority Voting ensemble.

Table 5.8: Overall Performance Comparison of All Sentiment Classification Models

Category	Model	Acc (%)	Prec (%)	Recall (%)	F1 (%)
Machine Learning	TF-IDF + LR	77.29	77.37	77.51	77.16
	Word2Vec + LR	73.76	73.71	74.08	73.75
	FastText + LR	74.76	74.76	74.95	74.70
	Doc2Vec + LR	66.75	67.76	68.01	66.71
	Word2Vec + SVM	69.92	69.93	70.07	69.94
	FastText + SVM	71.45	71.51	71.74	71.48
	Char TF-IDF + SVM	76.40	76.32	76.62	76.29
	Hybrid TF-IDF + SVM	76.93	76.77	77.05	76.77
	Fusion (Char + FT + SVM)	76.46	76.39	76.62	76.29
Deep Learning	Word2Vec + BiLSTM	77.60	77.75	77.83	77.51
	FastText + BiLSTM	76.20	76.55	76.35	76.11
	Word2Vec + Transformer	77.82	77.84	78.07	77.71
	FastText + Transformer	77.86	77.80	78.04	77.72
Pretrained Transformers	XLM-RoBERTa	79.07	78.88	79.09	78.85
	AraBERT	80.49	80.45	80.62	80.30
	MARBERT	81.56	81.27	81.41	81.25
	DziriBERT	**82.25**	**82.05**	**82.26**	**82.12**
Ensemble	**Hard Majority Voting**	**86.42**	**86.27**	**86.46**	**86.36**

As shown in Table 5.8, the machine learning models provided strong baseline performance, with TF-IDF + Logistic Regression achieving the highest accuracy among them at 77.29%. Character-based representations also produced competitive results, demonstrating that lexical and subword information plays an important role in handling the spelling variations and mixed writing styles commonly found in the Algerian dialect.

The deep learning models slightly improved upon the traditional machine learning approaches. Among them, the Transformer-based architectures performed better than the BiLSTM models, although the performance gap was relatively small. This indicates that combining pretrained word embeddings with deep neural networks is effective for

sentiment classification, but the improvement remains limited.

Among the individual models, the pretrained transformer models achieved the highest performance. In particular, DziriBERT outperformed all other individual approaches, reaching an accuracy of 82.25%, a Macro Precision of 82.05%, a Macro Recall of 82.26%, and a Macro F1-score of 82.12%.

The proposed Hard Majority Voting ensemble achieved the highest overall performance, with an Accuracy of 86.42%, a Macro Precision of 86.27%, a Macro Recall of 86.46%, and a Macro F1-score of 86.36%. These results demonstrate that combining the predictions of multiple classifiers leads to more robust and reliable sentiment classification by reducing the impact of errors made by individual models.

Overall, the experimental results indicate that pretrained transformer models constitute the strongest individual classifiers for Algerian dialect sentiment classification, with DziriBERT achieving the best individual performance. However, the proposed Hard Majority Voting ensemble further improved these results, making it the best-performing approach among all the evaluated methods.

5.7 Conclusion

This chapter covered the setup and testing of the sentiment classification models built for the AlgeriaPulse corpus. We explored and compared different machine learning, deep learning, pretrained transformer, and Hard Majority Voting ensemble approaches using standard evaluation metrics. The results showed that pretrained transformer models achieved the best performance among the individual classifiers, with DziriBERT obtaining the highest scores. Our analysis also highlighted the effectiveness of contextualized language representations while showing the continuous challenges posed by sarcasm, implicit meaning, dialect expressions, and informal social media text. The proposed Hard Majority Voting ensemble achieved the best overall performance, demonstrating that combining the predictions of multiple classifiers can improve the robustness and reliability of sentiment classification. In short, these findings confirm that sentiment in Algerian dialect comments can be accurately classified using domain-specific language models and ensemble learning techniques. Moving forward from these results, the next chapter presents the design and implementation of the AlgeriaPulse Software-as-a-Service (SaaS) platform, which integrates the best-performing sentiment analysis approach into a practical web-based application.

Chapter 6

Architecture and Design of the AlgerianPulse Platform

Introduction

The previous chapters presented the methodology adopted in this work, the sentiment analysis models considered, and the experimental evaluation that guided the selection of the final sentiment analysis approach. In this chapter, we move from the research phase to the implementation phase by presenting the design and development of **AlgerianPulse**, the web platform developed to integrate the proposed Hard Majority Voting sentiment analysis approach into a complete and practical application.

We first present the overall system architecture, followed by the technologies and development tools adopted. We then present the system design through UML diagrams before discussing the implementation of the user interface and the integration of the Hard Majority Voting sentiment analysis approach within the platform.

6.1 System Architecture

This section presents the overall architecture of the AlgerianPulse platform. In our work, we implemented a modular three-tier architecture that separates the presentation, application, and inference layers. This design improves the maintainability and scalability of the platform while ensuring efficient communication between its different components. The architecture consists of the following three layers:

- **Presentation Layer (Client Side):** We developed a responsive web interface using React, Vite, HTML5, CSS3, and JavaScript. This layer manages user interactions, including authentication, project management, social media account integration, performing sentiment analysis, and viewing the generated reports.
- **Application and Data Layer (Server Side):** We implemented the backend using FastAPI, which exposes RESTful APIs that process user requests, manage application logic, communicate with external services such as the YouTube Data API, and interact with Supabase for authentication and PostgreSQL-based data storage.
- **AI Inference Layer:** We integrated our sentiment analysis models into a Python-based inference engine. This layer performs sentiment prediction using DziriBERT, AraBERT, MARBERT, XLM-RoBERTa, Word2Vec Transformer,

FastText Transformer, and the Gemini API. The predictions generated by these models are then combined using the proposed **Hard Majority Voting** ensemble to produce the final sentiment classification.

Figure 6.1 illustrates the overall architecture of the AlgerianPulse platform.

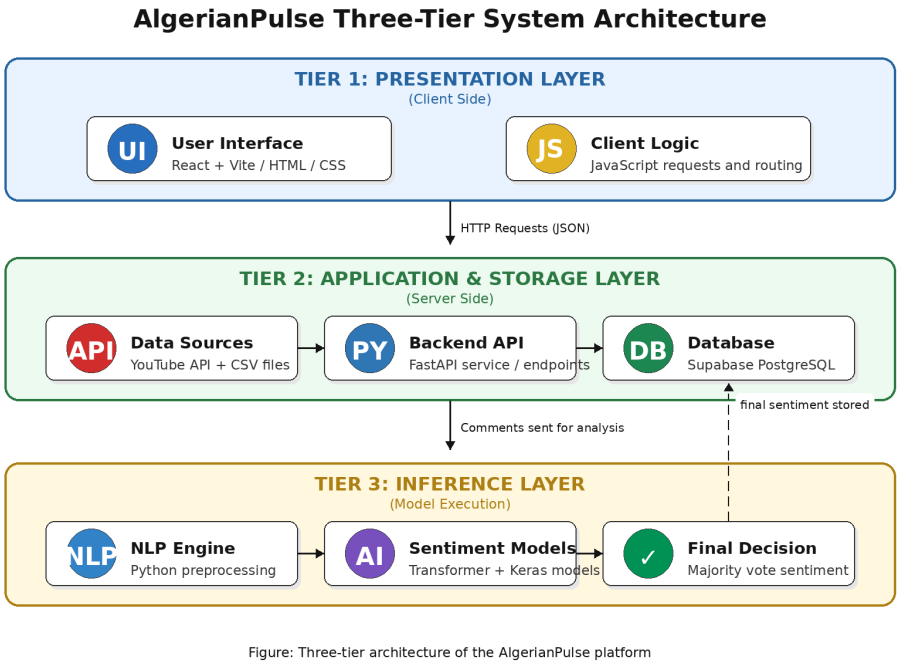


Figure 6.1: AlgerianPulse Three-Tier System Architecture (generated with AI assistance)

6.2 Development Environment

To develop the AlgerianPulse platform, we adopted a set of modern technologies and development tools that support the implementation of the frontend, backend, database, and AI components. Each technology was selected according to its role within the platform to ensure efficient development, scalability, and seamless integration between the different system layers. The following subsections present the technologies and tools used throughout the development process.

6.2.1 Frontend Technologies

We developed the frontend of the AlgerianPulse platform using React and Vite. React allowed us to build the user interface as reusable components, making the application easier to develop and maintain. Vite was used as the frontend build tool because it provides fast project initialization, efficient development, and quick updates during testing.

To design the user interface, we used HTML5, CSS3, and JavaScript. HTML5 was used to define the structure of the pages, CSS3 to create a responsive and modern interface, and JavaScript to implement the interactive features of the platform. We also used Lucide React and React Icons to provide consistent icons across the different pages of the application.

6.2.2 Backend Technologies

We implemented the backend of the AlgerianPulse platform using FastAPI, a Python framework for building RESTful APIs. FastAPI allowed us to process user requests, manage the application logic, communicate with external services, and connect the frontend with Supabase and the sentiment analysis models.

We also used Python as the main programming language because it provides extensive support for machine learning, data processing, and API development. This made it possible to integrate our sentiment analysis models into the platform and process comments efficiently. Supabase was used to manage authentication and store platform data through PostgreSQL, including users, projects, social accounts, posts, comments, predictions, and reports.

6.2.3 AI Technologies

To perform sentiment analysis, we integrated several AI models into the AlgerianPulse platform. These include DziriBERT, AraBERT, MARBERT, XLM-RoBERTa, Word2Vec Transformer, FastText Transformer, and the Gemini API. Each model predicts the sentiment of a given comment independently.

To improve the reliability of the final prediction, we combined the outputs of all models using the **proposed Hard Majority Voting** ensemble. The final sentiment is determined by the sentiment label predicted by the majority of the models and is then presented to the user through the platform.

6.2.4 Development Tools

Throughout the development of the AlgerianPulse platform, we used several tools to support coding, testing, version control, and system design. We used Visual Studio Code (VS Code) as the main development environment for both the frontend and backend. Node.js and npm were used to install and manage the required frontend packages and dependencies.

We also used Git and GitHub for version control, allowing us to track changes and manage the project efficiently. The AI models were trained and exported using Kaggle, while PlantUML was used to design the UML diagrams presented in this work.

6.3 UML Design

Before implementing the AlgerianPulse platform, we designed the main system functionalities using UML diagrams. These diagrams help describe how users interact with the platform and how the main software components are organized. In this section, we present the use case diagram and the class diagram of the system.

6.3.1 Use Case Diagram

The use case diagrams illustrate the main interactions between the actors and the AlgerianPulse platform. To improve readability, the functionalities are divided into two diagrams. Figure 6.2 presents the use cases available to the user, including authentication, project management, social account management, sentiment analysis, and report generation. Figure 6.3 presents the administrator use cases, which include user management, AI model management, activity monitoring, and platform configuration.

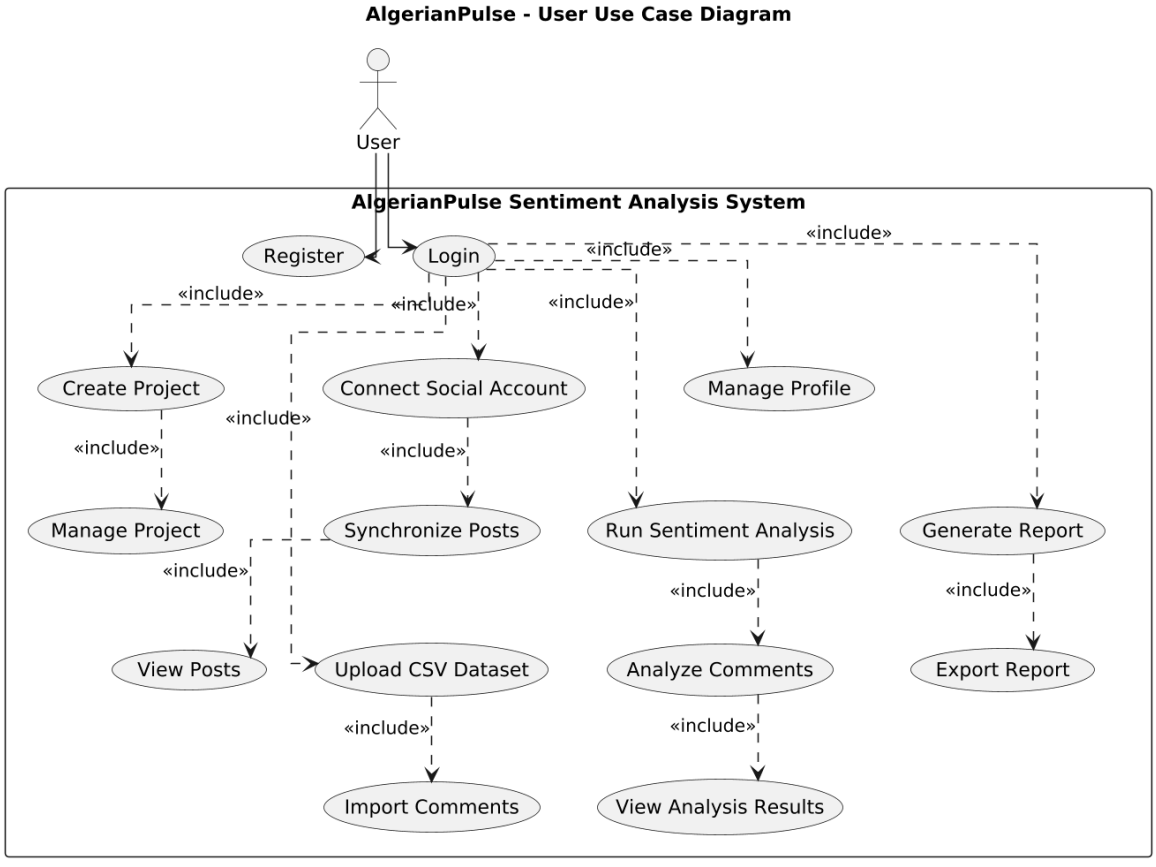


Figure 6.2: User use case diagram of the AlgerianPulse platform

6.3.2 Class Diagram

The class diagram represents the static structure of the AlgerianPulse platform and the relationships between its main entities. It shows the principal classes involved

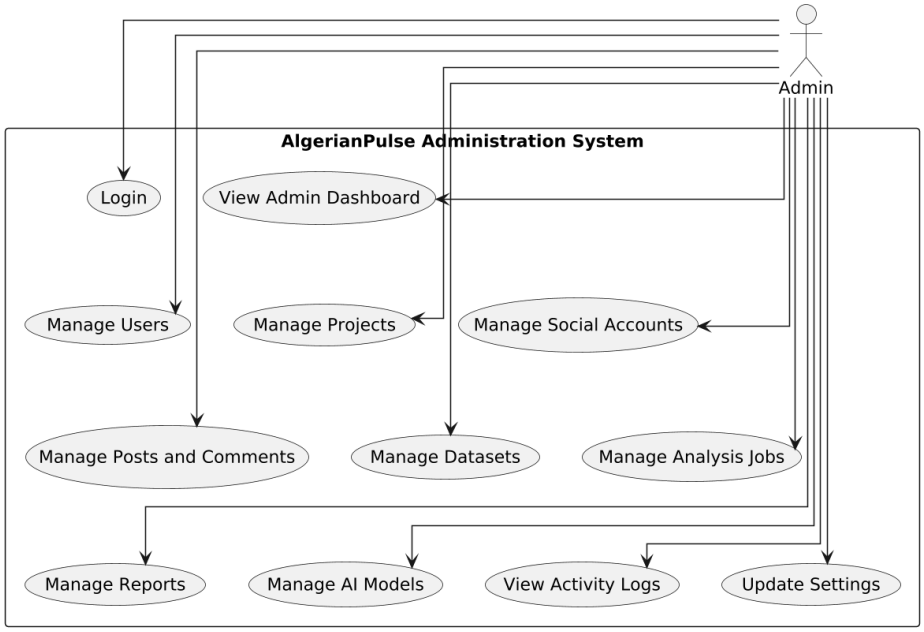


Figure 6.3: Administrator use case diagram of the AlgerianPulse platform

in the system, such as User, Admin, Project, SocialAccount, Post, Comment, AnalysisJob, ModelPrediction, PostAnalysis, and Report. This diagram helps clarify how the different components of the platform are related and how they support the main functionalities of the system.

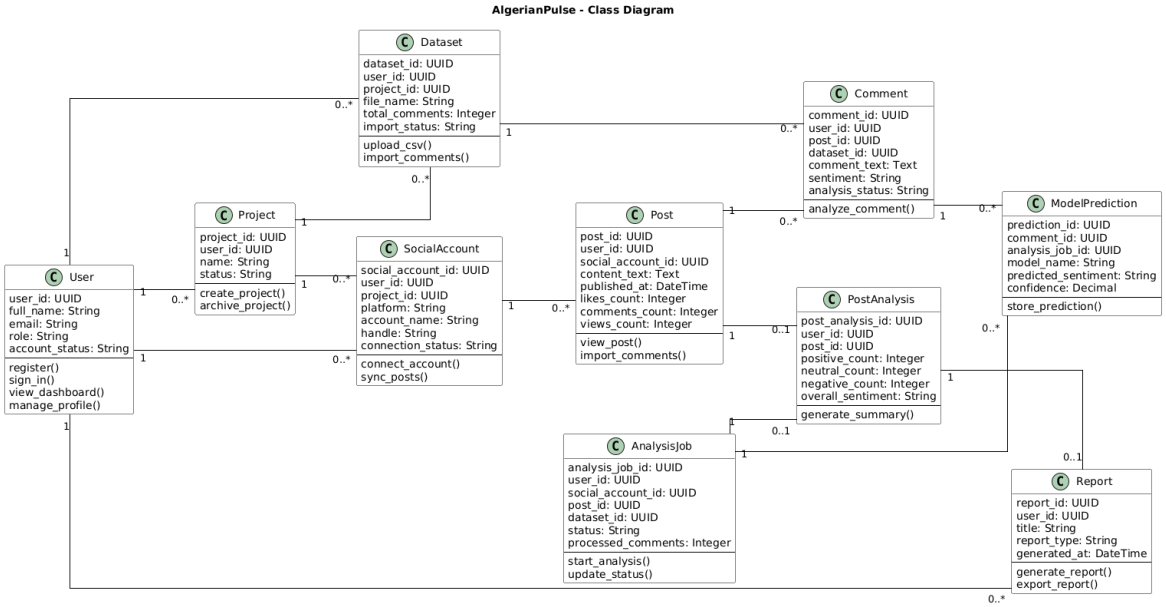


Figure 6.4: Class diagram of the AlgerianPulse platform

6.4 User Interface Implementation

After designing the system architecture and defining the UML models, we proceeded with the implementation of the AlgerianPulse platform. We developed a re-

sponsive web application that allows users to manage projects, connect social media accounts, perform sentiment analysis, and visualize the obtained results through an intuitive interface. In the following subsections, we present the main interfaces of the platform and describe their functionalities.

6.4.1 Homepage

The homepage is the entry point of the AlgerianPulse platform. We designed it to introduce the platform and present its main objectives and features in a clear and attractive way. It provides visitors with an overview of the platform's capabilities, including sentiment analysis, social media integration, AI assistance, trend tracking, and report generation. The homepage also allows users to navigate to the authentication page to sign in or create a new account and start using the platform.

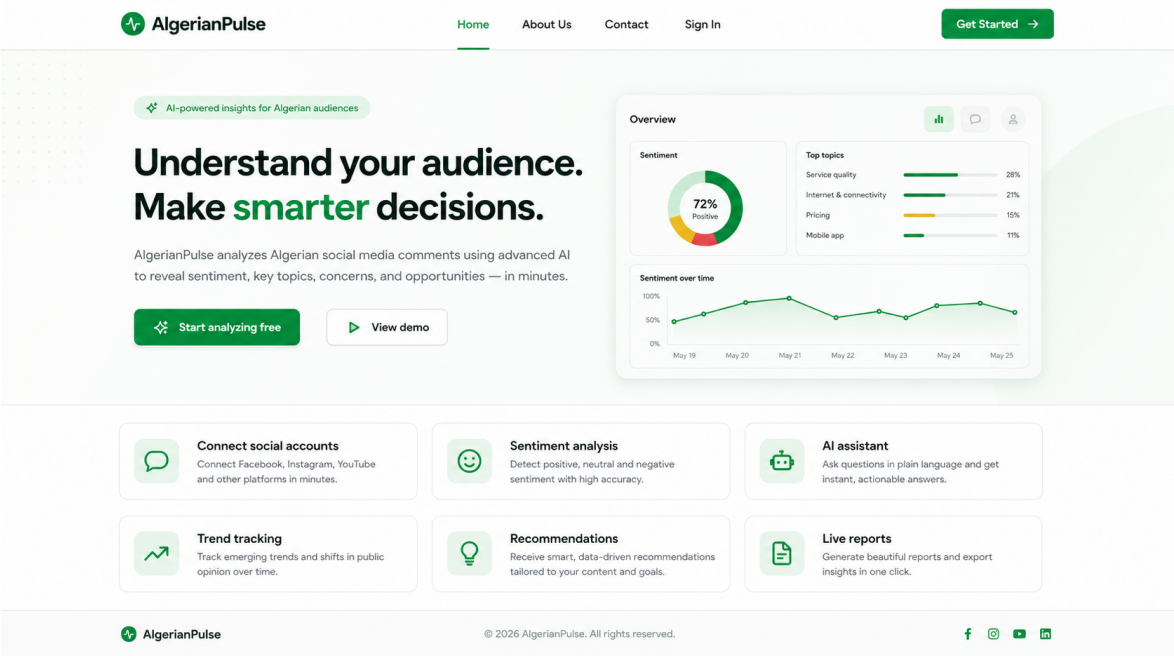


Figure 6.5: Homepage of the AlgerianPulse platform

6.4.2 User Authentication

The user authentication module allows users to securely access the AlgerianPulse platform. It provides two main functionalities: user registration and user login. New users can create an account by providing the required information, while existing users can authenticate using their email address and password to access their workspace. This module ensures that only authorized users can access the platform and its different services.

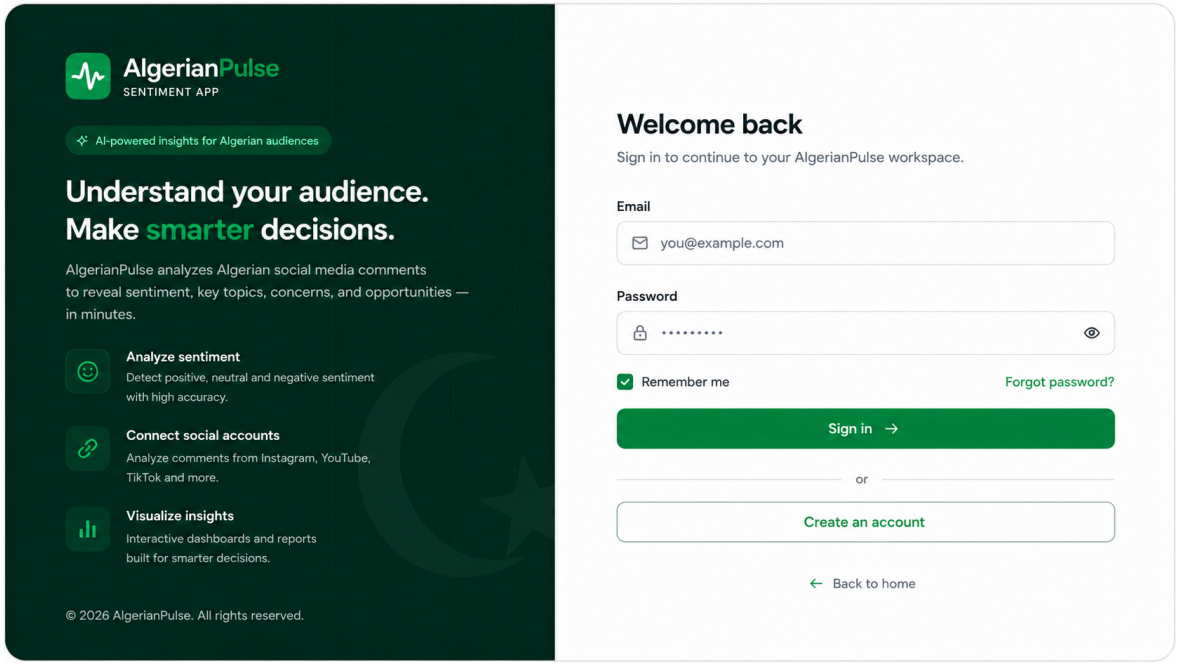


Figure 6.6: User authentication interface of the AlgerianPulse platform

6.4.3 User Dashboard

The user dashboard provides an overview of the user's activity within the AlgerianPulse platform. It summarizes the main statistics, including the number of connected social accounts, imported posts, analyzed comments, generated reports, and sentiment analysis results. The dashboard also displays interactive visualizations that help users monitor sentiment distribution and analysis trends, allowing them to quickly understand the overall performance of their projects based on the final predictions generated by the proposed Hard Majority Voting ensemble.

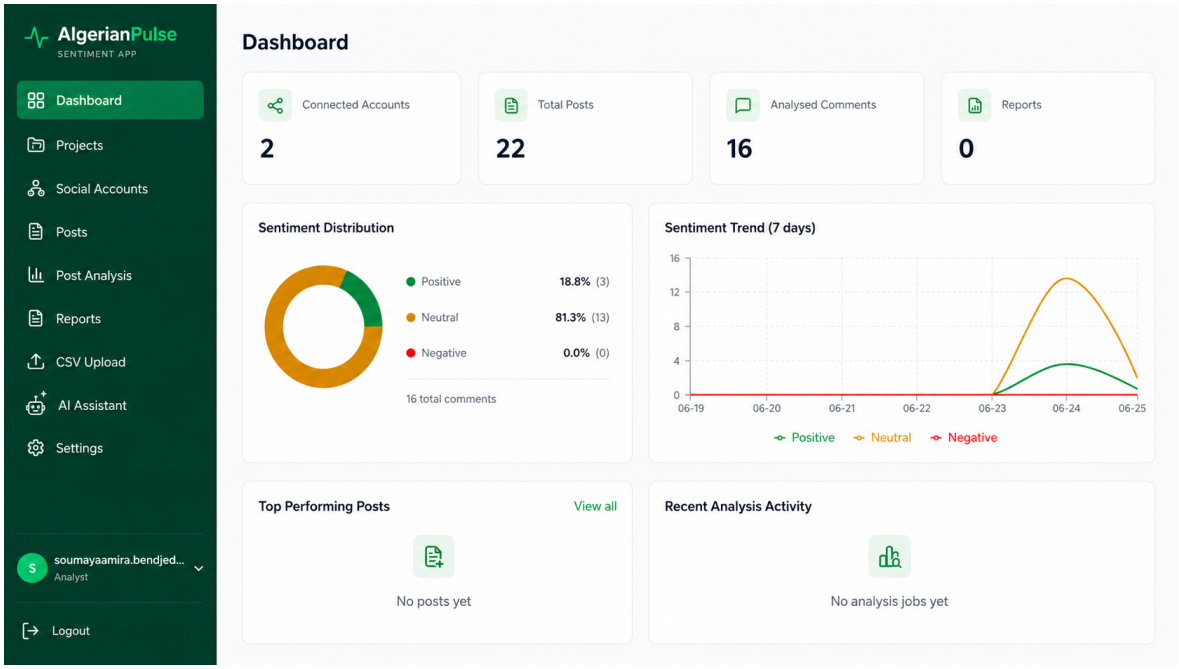


Figure 6.7: User dashboard of the AlgerianPulse platform

6.4.4 Social Account Management

The Social Account Management interface allows users to connect and manage their social media accounts within the AlgerianPulse platform. The current implementation supports YouTube, while the interface is designed to facilitate the integration of additional platforms such as Facebook, Instagram, and TikTok in future versions. After connecting an account, users can view the list of linked accounts together with their account name, identifier, connection date, and current status. The interface also provides options to add new accounts or remove existing ones, allowing users to efficiently organize the social media sources that will be used for sentiment analysis.

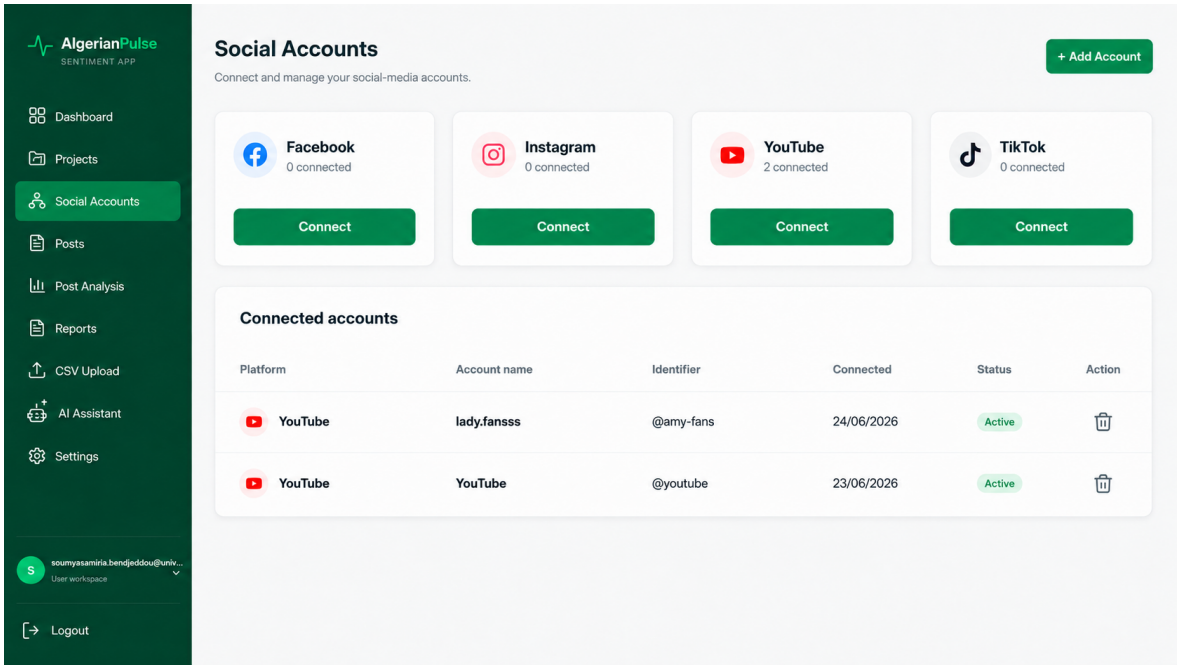


Figure 6.8: Social Account Management Interface

6.4.5 Posts Management

The Posts Management interface displays the posts imported from the user's connected social media accounts. Users can browse all imported posts, search for specific posts, and filter them according to the social media platform. For each post, the interface presents general information together with basic engagement statistics. It also provides direct actions to view the complete post, import its comments, or perform sentiment analysis. This interface serves as the main entry point for selecting the content that will be analyzed by the platform.

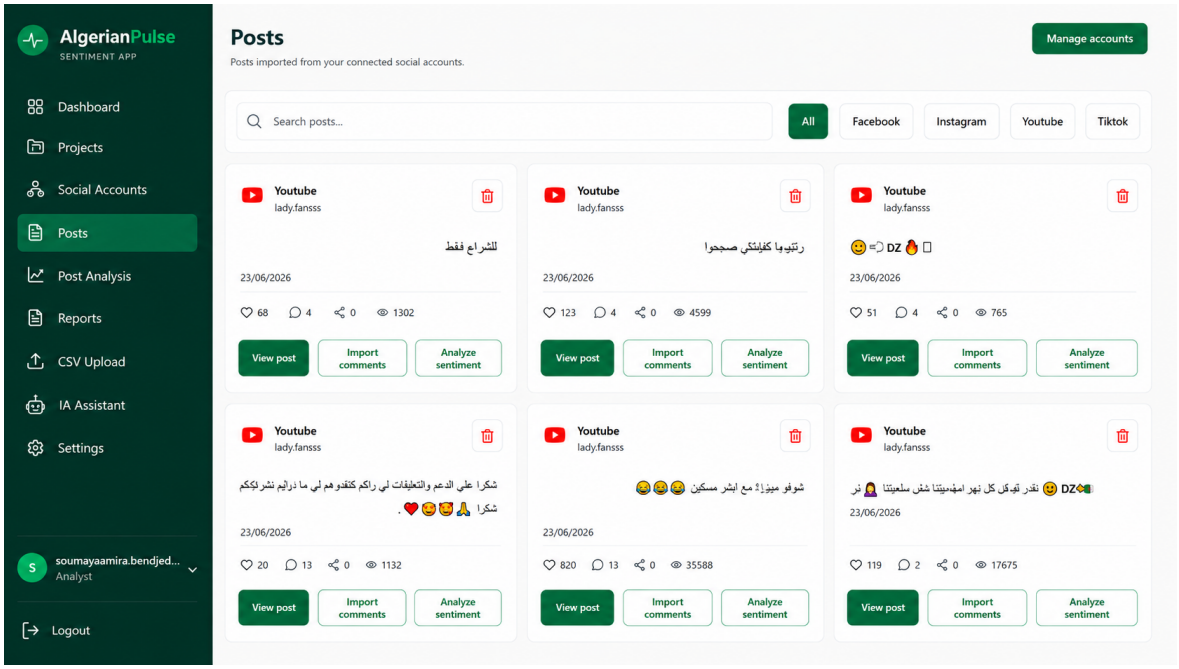


Figure 6.9: Posts Management Interface

6.4.6 CSV Dataset Upload

Besides importing comments from social media platforms, AlgerianPulse also supports uploading local CSV datasets. This feature is intended for users who already possess datasets collected from external sources or previous studies. Users simply upload a CSV file containing comments, after which the platform processes the data and makes it available for sentiment analysis using the integrated AI models. This functionality increases the flexibility of the platform by supporting both online and offline data sources.

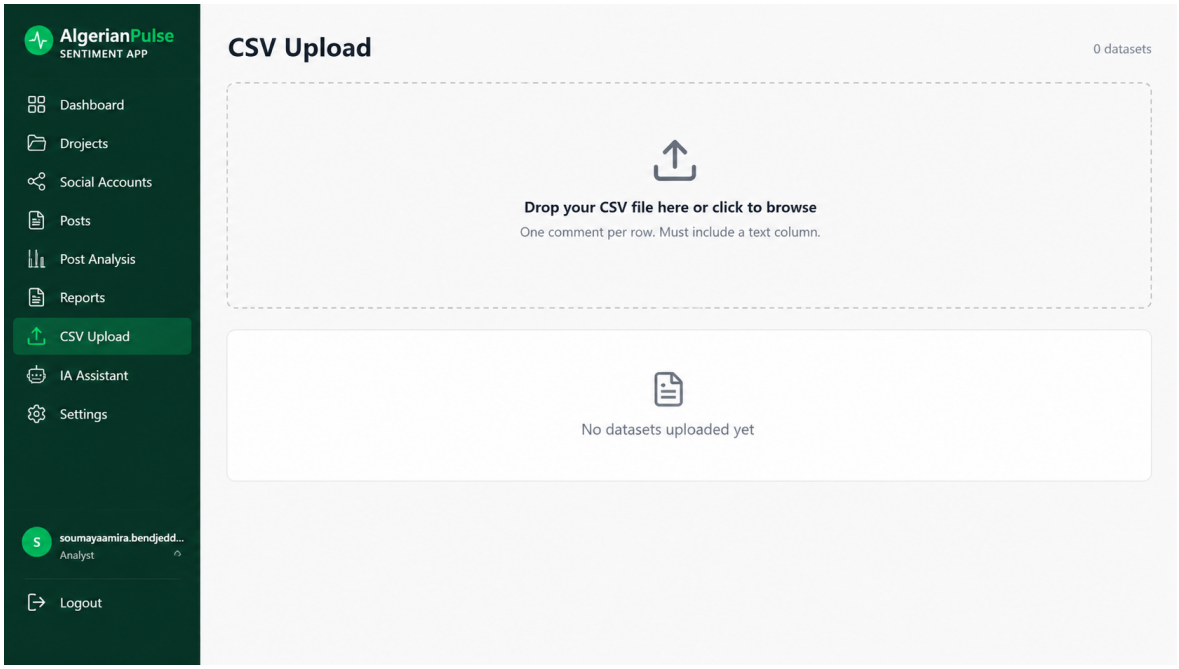


Figure 6.10: CSV Dataset Upload Interface

6.4.7 Post Analysis

The Sentiment Analysis interface presents the results produced by the integrated AI models after analyzing the imported comments. For each analyzed comment, the platform displays the final predicted sentiment together with summary statistics indicating the number of positive, neutral, and negative comments. This enables users to quickly evaluate the overall public opinion associated with the selected social media content.



Figure 6.11: Sentiment Analysis Interface

6.4.8 Report Generation

The Report Generation interface enables users to generate sentiment analysis reports for their projects. Users select the desired project together with a date range, after which the platform compiles the analysis results into a structured report. The generated report can then be exported for documentation, presentation, or further analysis, providing an easy way to share the obtained insights.

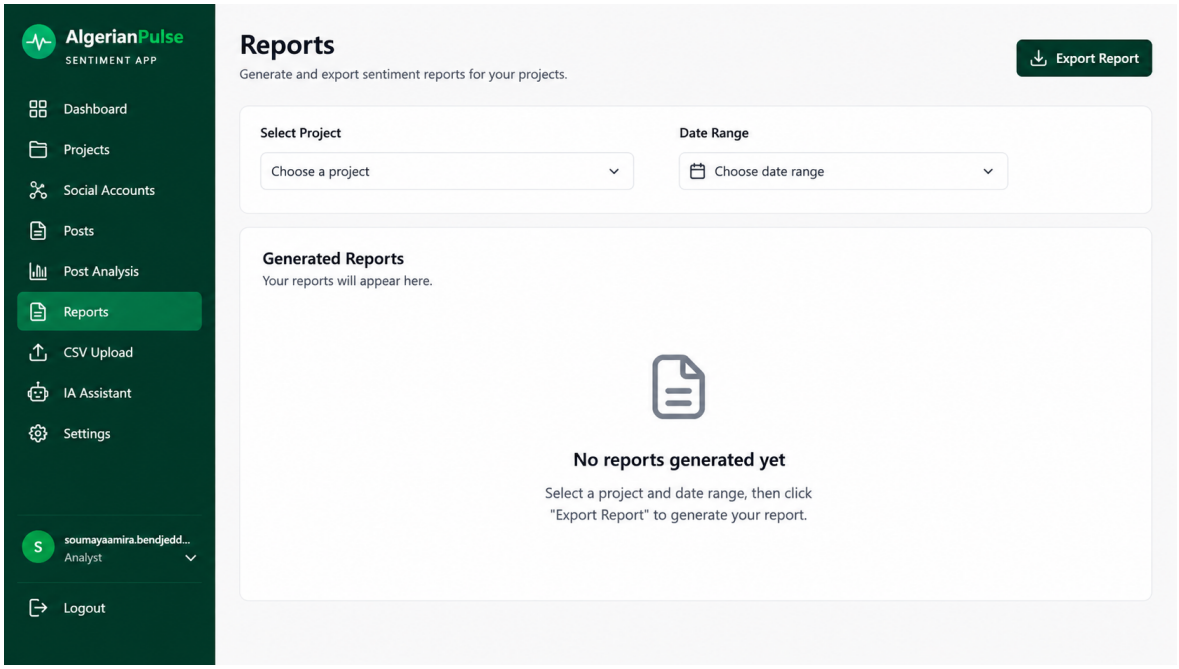


Figure 6.12: Report Generation Interface

6.5 Administrator Dashboard

The Administrator Dashboard provides an overview of the entire AlgerianPulse platform from the administrator's perspective. It summarizes important platform statistics, including the number of users, projects, datasets, social media accounts, reports, and analysis jobs. In addition, it offers access to administrative functionalities such as user management, AI model management, activity monitoring, and platform configuration, allowing administrators to efficiently supervise and maintain the system.

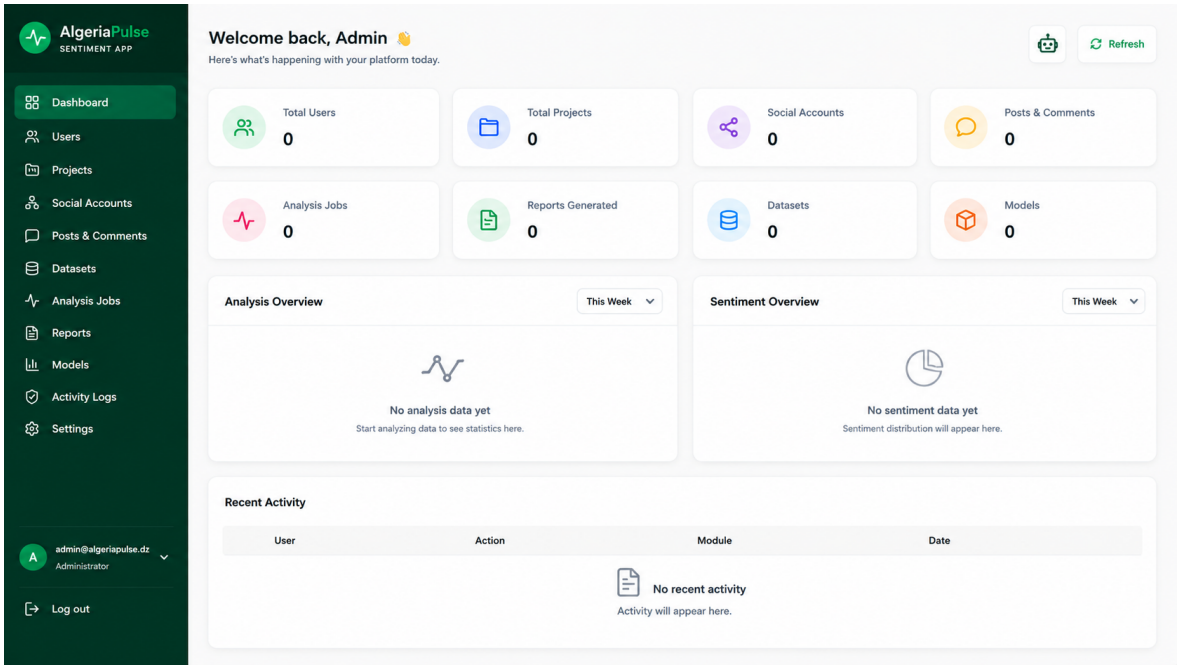


Figure 6.13: Administrator Dashboard

6.6 Integration of the Sentiment Analysis Models

After training and selecting the best-performing models presented in the previous chapter, they were integrated into the AlgerianPulse platform through the backend inference pipeline. Whenever a user initiates sentiment analysis, the backend retrieves the selected comments and forwards each comment to all integrated sentiment analysis models. Each model performs inference independently and returns a predicted sentiment together with its confidence score. The individual predictions are then combined using the proposed **Hard Majority Voting** ensemble to determine the final sentiment displayed by the platform.

6.6.1 AI Model Deployment

The trained sentiment analysis models were exported after the training phase and integrated into the FastAPI backend as independent inference modules. During server initialization, the models are loaded into memory and remain available to process prediction requests efficiently.

For every input comment, the backend sends the same text to each integrated model, namely DziriBERT, AraBERT, MARBERT, XLM-RoBERTa, the Word2Vec Transformer, and the FastText Transformer. Each model processes the comment independently and returns a predicted sentiment label together with its confidence score. This modular architecture allows the platform to evaluate the predictions of all models before producing the final sentiment classification.

6.6.2 Gemini API Integration

Unlike the locally deployed models, Gemini is accessed through Google's cloud API. For each analyzed comment, the backend sends an authenticated request containing the comment text to the Gemini API and receives a sentiment prediction as the response. The returned prediction is treated in the same manner as the predictions generated by the local models and becomes part of the proposed Hard Majority Voting ensemble.

Integrating Gemini provides an additional large language model that complements the transformer-based classifiers and contributes to improving the robustness of the final sentiment prediction.

6.6.3 Hard Majority Voting Integration

After collecting the predictions generated by all integrated models, the backend applies the proposed **Hard Majority Voting** ensemble. Each model contributes one vote corresponding to its predicted sentiment class. The backend counts the votes for the Positive, Neutral, and Negative classes, and the sentiment receiving the highest number of votes is selected as the final prediction presented to the user.

6.6.4 Prediction Storage and Visualization

Once the final sentiment has been determined, the backend stores the prediction results in the Supabase database together with the associated comment and analysis metadata. These stored results are subsequently retrieved by the frontend to update the dashboard, display the analyzed comments, generate sentiment statistics, and produce analytical reports. Storing the predictions also avoids repeating the inference process when users revisit previously analyzed posts.

6.7 Conclusion

This chapter presented the design and implementation of the AlgerianPulse platform. We first introduced the overall system architecture and the development environment used to build the platform. We then described the UML design through the use case and class diagrams. After that, we presented the main user and administrator interfaces of the application. Finally, we explained how the sentiment analysis models, including Gemini, were integrated into the backend and combined using the proposed Hard Majority Voting ensemble to produce the final sentiment prediction.

This implementation demonstrates the practical deployment of the proposed sentiment analysis framework within a complete web-based application. It also provides a solid foundation for real-world sentiment analysis on Algerian social media data. This leads directly into the **General Conclusion**, which summarizes the main contributions of this work, discusses its broader impact and current limitations, and outlines future directions for extending and improving the AlgerianPulse platform.

General Conclusion

This dissertation presented a complete study on sentiment analysis of Algerian Darija comments collected from social media platforms. The main objective was to build and evaluate sentiment classification models for Algerian dialect, which remains a low-resource language variety due to the lack of annotated datasets, linguistic resources, and standardized writing conventions.

Throughout this work, we first discussed the challenges of low-resource languages and the specific linguistic characteristics of Algerian Darija, including spelling variation, code-switching, informal expressions, Arabizi writing, emojis, and the absence of a standard orthography. We then reviewed the main text representation and classification methods used in Natural Language Processing, including TF-IDF, Word2Vec, FastText, traditional machine learning classifiers, deep learning models, and pretrained transformer-based models. In addition, we presented the main concepts of sentiment analysis and the difficulties of applying it to Algerian dialectal content.

The practical part of this work focused on the construction of the AlgeriaPulse corpus. The dataset was prepared through several stages, including data collection, cleaning, annotation, integration, preprocessing, and balancing. The comments were classified into three sentiment classes: positive, negative, and neutral. Several models were then trained and evaluated using standard metrics such as accuracy, precision, recall, and Macro F1-score. Among the individual classifiers, DziriBERT achieved the best performance, while the proposed Hard Majority Voting ensemble obtained the highest overall results by combining the predictions of multiple models, demonstrating improved robustness and reliability for sentiment classification.

Finally, this work included the implementation of the AlgerianPulse platform, a prototype system designed to support sentiment analysis of Algerian social media comments. The platform provides a practical interface for managing projects, datasets, analyses, and reports while integrating the proposed Hard Majority Voting ensemble into a complete application environment.

Despite the obtained results, several limitations remain. The Algerian dialect is highly diverse across regions, and online comments often contain noisy, ambiguous, and mixed-language content. In addition, the lack of large, high-quality annotated corpora limits the generalization ability of sentiment analysis models. Future work may focus on expanding the dataset, improving annotation quality, including more Algerian dialect varieties, exploring larger language models and ensemble learning techniques, and enhancing the deployed platform with more advanced analysis and visualization features.

Future Perspectives

Several future improvements can be considered to extend this work. First, the AlgeriaPulse corpus can be expanded by collecting more comments from different Algerian regions and social media platforms in order to improve the diversity and representativeness of the dataset. Second, the annotation process can be strengthened by involving more annotators and applying stricter agreement measures to improve the reliability of sentiment labels.

Another important direction is to improve the handling of Algerian dialect specificities, such as code-switching, Arabizi writing, emojis, slang expressions, and regional vocabulary. Future work may also explore larger pretrained language models, domain-adapted transformers, improved ensemble learning strategies, and fine-tuning approaches specifically designed for Algerian Arabic. In addition, multimodal sentiment analysis could be investigated by combining text with images, videos, or audio content from social media.

From the application side, the AlgeriaPulse platform can be further improved by adding more advanced visualization tools, real-time social media data collection, automatic report generation, user management features, online payment integration, and more flexible dashboard options. These improvements would make the system more useful for researchers, companies, and institutions interested in analyzing public opinion in the Algerian context.

References

[1] Tafa TO, Hashim SZM, Othman MS, etal. Machine translation performance for low-resource languages: a systematic literature review. IEEE Access. 2025;13:72486-72505. [doi: 10.1109/ACCESS.2025.3562918]

[2] I. Ezeani, P. Rayson, I. Onyenwe, C. Uchechukwu, and M. Hepple, "Igbo-English Machine Translation: An Evaluation Benchmark," *arXiv preprint*, arXiv:2004.00648, 2020.

[3] Juan Prieto, Cristian Martinez, Melissa Robles, Alberto Moreno, Sara Palacios, and Rubén Manrique. 2024. Translation systems for low-resource Colombian Indigenous languages, a first step towards cultural preservation. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pages 7–14, Mexico City, Mexico. Association for Computational Linguistics.[doi: 10.18653/v1/2024.americasnlp-1.2]

[4] K. Ghosh, S. Saha, T. Mandl, and S. Modha, "Findings from shared tasks on hate speech detection: Performance patterns for low-resource languages," *Pattern Recognition Letters*, vol. 199, pp. 303–309, Jan. 2026, doi: https://doi.org/10.1016/j.patrec.2025.09.004.

[5] I. Inuwa-Dutse, "NaijaNLP: A Survey of Nigerian Low-Resource Languages," *arXiv preprint*, arXiv:2502.19784, 2025.

[6] P. Pakray, A. Gelbukh, and S. Bandyopadhyay, "Natural language processing applications for low-resource languages," *Natural Language Processing*, vol. 31, pp. 183–197, Feb. 2025, doi:10.1017/nlp.2024.33.

[7] A. Soumeur, M. Mokdadi, A. Guessoum, and A. Daoud, "Sentiment analysis of users on social networks: Overcoming the challenge of the loose usages of the Algerian dialect," in Proc. 4th Int. Conf. Arabic Comput. Linguistics (ACLing), Dubai, UAE, Nov. 2018.

[8] I. Guellil, A. Adeel, F. Azouaou, F. Benali, A.-E. Hachani, K. Dashtipour, M. Gogate, C. Ieracitano, R. Kashani, and A. Hussain, "A semi-supervised approach for sentiment analysis of Arab(ic+izi) messages: Application to the Algerian dialect," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 212–225, 2021, doi: 10.1515/jisys-2020-0203.

[9] S. Harrat, K. Meftouh, M. Abbas, W.-K. Hidouci, and K. Smaïli, "An Algerian dialect: Study and resources," HAL preprint, 2017. https://hal.science/hal-01297415v1

[10] H. Saadane and N. Habash, "A conventional orthography for Algerian Arabic," in Proc. 2nd Workshop on Arabic Natural Language Processing, 2015, pp. 69–79.

[11] I. Guellil, F. Azouaou, H. Saâdane, and N. Semmar, “Une approche fondée sur les lexiques d’analyse de sentiments du dialecte algérien,” 2016. (Technical report / conference paper)

[12] Ouchene, L., Bessou, S. Benchmarking sentiment analysis of algerian arabic dialect on X (Twitter). J. King Saud Univ. Comput. Inf. Sci. **37**, 358 (2025). https://doi.org/10.1007/s44443-025-00295-w

[13] Ouchene, L., Bessou, S. AlgVec: A word embedding model for the algerian dialect in arabic and arabizi. J. King Saud Univ. Comput. Inf. Sci. **38**, 20 (2026). https://doi.org/10.1007/s44443-025-00407-6

[14] Y. Matrane, F. Benabbou, and N. Sael, ”A systematic literature review of Arabic dialect sentiment analysis,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 6, Jun. 2023, Art. no. 101570, doi: 10.1016/j.jksuci.2023.101570.

[15] R. Patil, S. Boit, V. Gudivada, and J. Nandigam, ”A survey of text representation and embedding techniques in NLP,” *IEEE Access*, vol. 11, pp. 33661–33689, 2023, doi: 10.1109/ACCESS.2023.3266377.

[16] Sutriawan, Muljono, Khairunnisa, Z. Alamin, T. A. Lorosae, and S. Ramadhan, ”Improving performance sentiment movie review classification using hybrid feature TF-IDF, N-gram, information gain and Support Vector Machine,” *Math. Model. Eng. Probl.*, vol. 11, no. 2, pp. 375–384, Feb. 2024, doi: 10.18280/mmep.110209.

[17] A. C. Mazari and A. Djeffal, ”Sentiment analysis of Algerian dialect using machine learning and deep learning with Word2Vec,” *Informatica*, vol. 46, no. 6, pp. 67–78, 2022, doi: 10.31449/inf.v46i6.3340.

[18] H. Alkaabi, A. K. Jasim, and A. Darroudi, ”From static to contextual: A survey of embedding advances in NLP,” *PERFECT: J. Smart Algorithms*, vol. 2, no. 2, pp. 57–66, Jul. 2025, doi: 10.62671/perfect.v2i2.77.

[19] K. Das, Kamlish, and F. Abid, ”Advancements in word embeddings: A comprehensive survey and analysis,” *Proc. Pak. Acad. Sci. A: Phys. Comput. Sci.*, vol. 61, no. 3, pp. 227–245, 2024, doi: 10.53560/PPASA(61-3)842.

[20] K. Babić, S. Martinčić-Ipšić, and A. Meštrović, ”Survey of neural text representation models,” *Information*, vol. 11, no. 11, Art. no. 511, 2020, doi: 10.3390/info11110511.

[21] A. Madasu and E. Sivasankar, ”A study of feature extraction techniques for sentiment analysis,” arXiv preprint arXiv:1906.01573, Jun. 2019.

[22] N. A. Smith, ”Contextual word representations: A contextual introduction,” Paul G. Allen School of Computer Science and Engineering, Univ. of Washington, Seattle, WA, USA, & Allen Inst. for AI, Tech. Rep., Apr. 2020.

[23] M. I. Ali and S. Supatmi, ”Comparative analysis of Logistic Regression and Support Vector Machine for sentiment classification of Shopee app reviews on the Play Store,” in *Proc. Int. Conf. Informatics, Eng., Sci. Technol. (INCITEST)*, vol. 1,

2025, pp. 50–61, doi: 10.34010/incitest.v1i.829.

[24] R. D. Cahyani and P. T. Prasetyaningrum, "Sentiment analysis of user reviews for AI applications: Evaluating SVM, Logistic Regression, and Random Forest," *J. Inf. Syst. Inform.*, vol. 8, no. 1, pp. 1–27, Feb. 2026, doi: 10.63158/journalisi.v8i1.1366.

[25] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[26] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.

[27] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," arXiv preprint arXiv:1508.01991, 2015.

[28] R. Panchendrarajan and A. Amaresan, "Bidirectional LSTM-CRF for Named Entity Recognition," in *Proc. 32nd Pac. Asia Conf. Lang., Inf. Comput. (PACLIC 32)*, 2018, pp. 221–230.

[29] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[30] Q. Fournier, G. M. Caron, and D. Aloise, "A practical survey on faster and lighter transformers," *ACM Comput. Surv.*, vol. 55, no. 14S, pp. 1–40, Jul. 2023, doi: 10.1145/3578933.

[31] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv preprint arXiv:1607.06450, 2016.

[32] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," in *Proc. 4th Workshop Open-Source Arabic Corpora Process. Tools (OSACT)*, Marseille, France, 2020, pp. 9–15.

[33] A. Abdaoui, M. Oussalah, M. Berrimi, and A. Moussaoui, "DziriBERT: A pre-trained language model for the Algerian dialect," arXiv preprint arXiv:2109.12346, Dec. 2022.

[34] H. R. A. Drous, H. M. S. Barakat, and A. A. B. Atta, "Fine-tuning MARBERT for sentiment analysis in Jordanian Arabic dialects using a synthetic dialectal corpus," *J. Comput. Sci. Technol. Stud.*, vol. 6, no. 1, pp. 1–10, 2024, doi: 10.32996/jcsts.2024.6.1.1.

[35] A. Conneau *et al.*, "Unsupervised cross-lingual representation learning at scale," arXiv preprint arXiv:1911.02116, 2020.

[36] Ravi, K., and Ravi, V. (2015). "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications." *Knowledge-Based Systems*, 89, 14–46. https://www.sciencedirect.com/science/article/abs/pii/S0950705115002336

[37] B. Liu, "Sentiment Analysis and Subjectivity," *Handbook of Natural Language Processing*, 2nd ed., 2010.

[38] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, Cambridge University Press (Lecture Excerpts/Textbook Data).

[39] Medhat, W., Hassan, A., and Korashy, H. (2014). "Sentiment analysis algorithms and applications: A survey." *Ain Shams Engineering Journal*, 5(4), 1093–1113. https://www.sciencedirect.com/science/article/pii/S2090447914000550

[40] Dalal, M. K., & Zaveri, M. A. (2014). "Opinion Mining from Online User Reviews Using Fuzzy Linguistic Hedges." *Applied Computational Intelligence and Soft Computing*, 2014. (Note: This reference corresponds to your third attached PDF document on fuzzy-based online user reviews opinion mining).

[41] Dalal, M. K., & Zaveri, M. A. (2014). "Opinion Mining from Online User Reviews Using Fuzzy Linguistic Hedges." *Applied Computational Intelligence and Soft Computing*, 2014, Article ID 735942, 9 pages. https://doi.org/10.1155/2014/735942

[42] I. Guellil, M. Mendoza, and F. Azouaou, "Arabic dialect sentiment analysis with ZERO effort. Case study: Algerian dialect," *Inteligencia Artificial*, vol. 23, no. 65, pp. 124–135, 2020. https://doi.org/10.4114/intartif.vol23iss65pp124-135

[43] A. C. Mazari and A. Djeffal, "Sentiment Analysis of Algerian Dialect Using Machine Learning and Deep Learning with Word2vec," 2022.

[44] L. Ouchene and S. Bessou, "Benchmarking sentiment analysis of Algerian Arabic dialect on X (Twitter)," *Journal of King Saud University – Computer and Information Sciences*, vol. 37, art. 358, 2025. https://doi.org/10.1007/s44443-025-00295-w

[45] I. Guellil, "Arabic dialect sentiment analysis with ZERO effort. Case study: Algerian dialect," *Inteligencia Artificial*, vol. 23, no. 65, pp. 124–135, 2020.

[46] L. Mazari and A. Kheddar, "A multi-label annotated Algerian dialect dataset for online hate speech and offensive language detection," Under Review, 2023.

[47] A. Abdaoui, M. Oussalah, M. Berrimi, and A. Moussaoui, "DziriBERT: A Pre-trained Language Model for the Algerian Dialect," in *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, 2022, arXiv:2109.12346v3.

[48] M. Lanasri et al., "Sentiment classification for Algerian dialect using fine-tuned pre-trained models," *Hugging Face Repository*, 2023.

[49] Grandini, M., Bagli, E., & Visani, G. (2020). "Metrics for Multi-Class Classification: An Overview." arXiv:2008.05756. https://arxiv.org/pdf/2008.05756

[50] H. N. Mhaskar and S. V. Pereverzyev, "New bounds on the accuracy of majority voting for multi-class classification," arXiv preprint arXiv:2309.09564, pp. 1–15, Sep. 2023. [Online]. Available: https://arxiv.org.