



# **Enhancing Arabic Offensive Tweet Classification: An Ensemble Approach Integrating AraBERT, Neural Networks, and LSTM Models**

**by**

**AHLAM MOHAMMAD WAHDAN**

**A thesis submitted in partial fulfilment  
of the requirements for the degree of  
DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE  
at  
The British University in Dubai**

**October 2023**

## DECLARATION

I warrant that the content of this research is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that a copy of my research will be deposited in the University Library for permanent retention.

I hereby agree that the contents of this dissertation for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study, or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make a digital copy available in the institutional repository.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my thesis for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.

A handwritten signature in black ink, consisting of several overlapping loops and a final flourish.

---

Signature of the student

## **COPYRIGHT AND INFORMATION TO USERS**

The author whose copyright is declared on the title page of the work has granted to the British University in Dubai the right to lend his/her research work to users of its library and as an open source and to make partial or single copies for educational and research use.

The author has also granted permission to the University to keep or make a digital copy for similar use and for the purpose of preservation of the work digitally.

Multiple copying of this work for scholarly purposes may be granted by either the author, the Registrar, or the Dean only.

Copying for financial gain shall only be allowed with the author's express permission.

Any use of this work in whole or in part shall respect the moral rights of the author to be acknowledged and to reflect in good faith and without detriment the meaning of the content, and the original authorship.

## **ABSTRACT**

This thesis addresses the crucial research problem of accurate detection and moderation of offensive language in Arabic text, considering the intricacies posed by the language's complex morphology, dialectal variations, orthographic ambiguity, orthographic noise, limited linguistic resources, and the necessity for comprehensive coverage of offensive language expressions. The research objectives are delineated through four key research questions. Firstly, the study aims to identify the existing research gaps in Arabic Text Classification (ATC) through an extensive and rigorous systematic literature review. The study adopts a scholarly and formal approach, aiming to identify the specific areas within ATC research that lack comprehensive exploration or exhibit inadequacies in existing knowledge. This endeavor is grounded in the rigorous analysis and synthesis of relevant academic literature, ensuring a meticulous examination of the current state of research in ATC. Secondly, it investigates the effects of employing novel pre-processing methods on the performance of Arabic Text Classification. Thirdly, the research endeavors to determine the most effective model for enhancing the accuracy of Arabic offensive text classification by introducing a novel approach using pre-trained models; AraBERT model in conjunction with fully connected neural networks (NN) and long short-term memory (LSTM) networks. Finally, the study evaluates the proposed model's ability to classify Arabic offensive text effectively. The research methodology consists of two integral parts, comprising dataset description, the proposed framework. The dataset description provides insights into the two datasets utilized, namely OSACT and Semeval. The framework elucidates the proposed model, which leverages a combination of pretrained models and neural networks, thereby achieving a high level of effectiveness in classifying Arabic offensive text. The model's performance is meticulously

assessed using various evaluation metrics, including accuracy and F1-macro score, and is compared against other classifier models. The research findings demonstrate the superiority of the proposed model over the baseline AraBERT model, with the proposed model achieving an accuracy of 0.870 compared to the baseline accuracy of 0.820, along with an F1-score of 0.853 compared to the baseline's 0.800. This emphasizes the model's exceptional capacity to accurately identify offensive content in Arabic text. The implications of this research extend to diverse domains and stakeholders, encompassing decision makers, developers, and policy makers. The insights garnered from the study can be instrumental in making informed decisions pertaining to the integration of Arabic text classification systems in various operational settings. By comprehending the proposed model's performance and efficacy, decision makers can assess its potential impact on optimizing processes such as information retrieval, content filtering, and sentiment analysis in Arabic text. In conclusion, this thesis contributes significantly to the existing literature by addressing the complexities associated with offensive language identification in Arabic text and introducing an innovative approach that integrates pretrained models with deep learning techniques and neural networks. The demonstrated effectiveness and superior performance of the proposed model underscore its potential for practical implementation in real-world scenarios, thereby bolstering the field of Arabic offensive text classification.

**Keywords:** Arabic offensive classification, preprocessing techniques, AraBERT, AraBERT preprocessing, ensemble methodology, deep learning, neural networks, LSTM, natural language processing, emoji interpretation.

## الخلاصة

تتناول هذه الرسالة المشكلة البحثية الحرجة للكشف والتحكم الدقيق في اللغة المسيئة في النصوص العربية، مع مراعاة التحديات و التعقيدات التي يتسبب فيها تعقيد المورفولوجيا اللغوية، وتباين اللهجات، والغموض الإملائي، والوضاء الإملائية، والموارد اللغوية المحدودة، وضرورة تغطية شاملة لتعابير اللغة المسيئة. تم تحديد أهداف البحث من خلال أربعة أسئلة بحثية رئيسية. أولاً، تهدف الدراسة إلى تحديد الفجوات البحثية الحالية في تصنيف النصوص باللغة العربية من خلال استعراض أدبي منهجي ودقيق وشامل. تتبنى الدراسة نهجاً علمياً ورسمياً، بهدف تحديد المجالات الخاصة داخل بحث تصنيف النصوص باللغة العربية التي تفتقر إلى استكشاف شامل أو تظهر نقائصاً في المعرفة الحالية. يقوم هذا الجهد على تحليل وتوليف دقيق للأدب الأكاديمي ذي الصلة، مما يضمن فحصاً دقيقاً للحالة الحالية للبحث في تصنيف النصوص باللغة العربية. ثانياً، يقوم البحث بدراسة تأثير استخدام أساليب معالجة جديدة على أداء تصنيف النصوص باللغة العربية. ثالثاً، يسعى البحث إلى تحديد النموذج الأكثر فعالية لزيادة دقة تصنيف النصوص المسيئة باللغة العربية من خلال استخدام نهج مجمع من اربيرت و الشبكات ذات الذاكرة القصيرة و الطويلة الأمد ،و الشبكات العصبية المتصلة بالكامل. و أخيراً يقوم البحث بتقييم قدرة النموذج المدمج المقترح على تصنيف التغريدات المسيئة باللغة العربية بفعالية. منهجية البحث تتكون من جزئين أساسيين، يتضمن الجزء الأول وصف مجموعة البيانات والجزء الثاني يتضمن الاطار المقترح للبحث. بالنسبة للجزء الأول وصف لمجموعة البيانات فهو يقدم رؤية حول البيانات المستخدمة لمجموعتي البيانات اوسكت و سيم ايفال. أما الجزء الثاني الذي يتضمن الاطار المقترح للبحث فهو يوضح النموذج المقترح الذي يستفيد من النماذج المدربة مسبقاً و الشبكات العصبية، مما يسمح بتحقيق مستوى عال من الفعالية في تصنيف النصوص المسيئة باللغة العربية. حيث يتم تقييم أداء النموذج المقترح بعناية باستخدام مقاييس متنوعة. حيث يتم استخدام مقاييس الدقة و الماكرو اف ١، وتتم مقارنته بنماذج متنوعة. تظهر نتائج البحث تفوق النموذج المقترح على نموذج اربيرت الأساسي باستخدام مقياس الدقة ب ٠,٨٧٣، للنموذج المقترح ، و ٠,٨٥٢ للنموذج الأساسي. و تفوق النموذج المقترح على نموذج اربيرت الأساسي باستخدام مقياس الماكرو اف ١ ب ٠,٨٥٣، للنموذج المقترح ، و ٠,٨٠٠ للنموذج الأساسي. وهذا يؤكد على قدرة النموذج المقترح الاستثنائية على التعرف بدقة على المحتوى الجارح في النصوص العربية. تمتد آثار هذا البحث إلى مجموعة متنوعة من المجالات وأصحاب المصلحة، بما في ذلك اتخاذ القرار، ومطوري البرمجيات، وصانعي السياسات. يمكن أن تكون الرؤى التي تم الحصول عليها من الدراسة مفيدة في اتخاذ قرارات مستنيرة تتعلق بدمج أنظمة تصنيف النصوص باللغة العربية في مختلف

الإعدادات التشغيلية. من خلال فهم أداء النموذج المقترح وفعاليته، يمكن لاتخاذ القرارات تقييم تأثيره المحتمل على تحسين العمليات مثل استرجاع المعلومات وتصفية المحتوى وتحليل المشاعر في النصوص العربية. في الختام، تسهم هذه الرسالة بشكل كبير في الأدب الموجود بالتعامل مع تعقيدات تعرف تحديد اللغة الجارحة في النصوص العربية وباقتراح نهج مبتكر يدمج النماذج المدربة مسبقًا مع تقنيات التعلم العميق والشبكات العصبية. الفعالية المظهرة والأداء الفائق للنموذج المقترح تسلط الضوء على إمكانياته للتنفيذ العملي في سياقات العمل الحقيقية، مما يعزز مجال تصنيف النصوص الجارحة باللغة العربية.

الكلمات الرئيسية: تصنيف النصوص الجارحة باللغة العربية، تقنيات المعالجة المسبقة، نموذج

، معالجة اللغة LSTM المسبقة، منهجية الانتقاء، التعلم العميق، الشبكات العصبية، AraBERT ، معالجةAraBERT

.الطبيعية، تفسير الإيموجي

## DEDICATION

I extend my dedication of this thesis to the individuals who have provided me with unwavering support, believed in my capabilities, and stood by me at every juncture of my academic journey. With deep affection, I dedicate this thesis to my parents, whose sacrifices and encouragement were my unwavering motivation. To my spouse, whose unwavering support and understanding made this journey possible, even though it meant taking on extra responsibilities at home, from daily chores to studying. To my daughters, who inspired me to persevere and set an example for them, even when it meant being a part due to my studies. I'm profoundly grateful for the sacrifices you've made and the strength you've shown during this time. Your unwavering love, belief, and support have been the driving force behind this achievement, and I'm deeply appreciative of everything you've done. To my sisters and brothers, who offered their continuous support and encouragement. To my friends, whose companionship and encouragement provided moments of respite during the challenging times. Your love and belief in me, coupled with the unwavering support of my family, have been the driving force behind this achievement, and I'm profoundly grateful for the strength you've provided throughout this journey.

## **ACKNOWLEDGEMENTS**

In the name of God, I begin this dedication with gratitude and praise for the countless blessings bestowed upon me. I would like to express my utmost respect and appreciation to my research 1st supervisor, Prof. Khaled Shaalan. His unwavering support and guidance have been invaluable throughout this journey. His exceptional vision, wisdom, motivation, and vast knowledge have profoundly inspired and encouraged me. Without his guidance, this thesis would not have come to fruition. I would also like to extend my sincere and heartfelt gratitude to my 2nd supervisor, Dr. Mostafa Al-Emran. His continuous encouragement, honesty, analytical thinking, and extensive experience have played a pivotal role in my success in accomplishing this task. I am truly grateful for his guidance and mentorship.

A special and profound thanks goes to my husband, who has stood by my side throughout this journey. He has endured the challenges posed by my fluctuating moods and demanding work schedule during weekends. His unwavering love and unwavering support have been the guiding light during the darkest moments. I am forever indebted to my parents for their unwavering encouragement, prayers, and boundless love. Additionally, my appreciation extends to my beloved daughters, who have been my pillars of strength. Their unwavering support has been a constant source of motivation and inspiration. To all those who have played a significant role in my academic journey and the completion of this thesis, I express my deepest gratitude and respect.

## TABLE OF CONTENTS

|                                                     |           |
|-----------------------------------------------------|-----------|
| <b>Chapter 1: Introduction.....</b>                 | <b>1</b>  |
| 1.1. Background .....                               | 1         |
| 1.2 Motivation .....                                | 8         |
| 1.3 Problem Statement .....                         | 11        |
| 1.4 Research Questions .....                        | 14        |
| 1.5 Research Aim and Objectives .....               | 15        |
| 1.6 Research Significance .....                     | 16        |
| 1.7 Research Structure.....                         | 19        |
| <b>Chapter 2: Technical Background.....</b>         | <b>20</b> |
| 2.1. Feature Extraction .....                       | 21        |
| 2.2. Classification Techniques .....                | 23        |
| 2.2.1. Manual and Statistical Techniques.....       | 23        |
| 2.2.2. Machine Learning Techniques .....            | 24        |
| 2.2.3. Neural Network .....                         | 25        |
| 2.3. Transformers .....                             | 29        |
| 2.4. Performance Evaluation .....                   | 35        |
| 2.5. Conclusion.....                                | 37        |
| <b>Chapter 3: Literature Review .....</b>           | <b>39</b> |
| 3.1. Overview .....                                 | 40        |
| 3.2. Arabic Text Classification.....                | 40        |
| 3.3. Offensive Language Identification Methods..... | 41        |

|                   |                                                                              |           |
|-------------------|------------------------------------------------------------------------------|-----------|
| 3.4.              | Systematic Literature Review of ATC .....                                    | 46        |
| 3.4.1.            | Inclusion and Exclusion Criteria .....                                       | 47        |
| 3.4.2.            | Search Process.....                                                          | 48        |
| 3.4.3.            | Quality Assessment .....                                                     | 50        |
| 3.4.4.            | Data Collection.....                                                         | 53        |
| 3.5.              | Systematic Review Results .....                                              | 54        |
| 3.5.1.            | Research Domains in Arabic Text Classification .....                         | 56        |
| 3.5.2.            | Proposed Solutions in Arabic Text Classification.....                        | 85        |
| 3.6.              | A Comparative Analysis of this Systematic Reviews and Previous Review .....  | 86        |
| 3.7.              | Systematic Review Discussion.....                                            | 91        |
| 3.8.              | Systematic Review Summary.....                                               | 94        |
| <b>Chapter 4:</b> | <b>Methodology .....</b>                                                     | <b>96</b> |
| 4.1.              | Dataset Description .....                                                    | 97        |
| 4.1.1.            | OSACT Dataset.....                                                           | 98        |
| 4.1.2.            | SemEval .....                                                                | 99        |
| 4.2.              | The Framework .....                                                          | 100       |
| 4.2.1.            | Preprocessing Module .....                                                   | 101       |
| 4.3.              | Classification Ensemble Module.....                                          | 113       |
| 4.3.1.            | AraBERT Pretraining.....                                                     | 113       |
| 4.3.2.            | Enhancements by Adding LSTM.....                                             | 114       |
| 4.3.3.            | Linear Layers.....                                                           | 115       |
| 4.4.              | Significance of Ensemble Approach in Combining AraBERT with the Custom Model | 116       |
| 4.5.              | Conclusion.....                                                              | 118       |

|                                                                     |            |
|---------------------------------------------------------------------|------------|
| <b>Chapter 5: Experiments and Results.....</b>                      | <b>121</b> |
| 5.1. Experimental Setup .....                                       | 122        |
| 5.1.1. Hardware Configuration.....                                  | 122        |
| 5.1.2 Software Configuration .....                                  | 122        |
| 5.2. Experimental Results.....                                      | 124        |
| 5.2.1. Preprocessing Results.....                                   | 124        |
| 5.2.2. Classification Results .....                                 | 127        |
| 5.2.3. Incorporating additional dataset .....                       | 131        |
| 5.2.4. Feature Extraction Techniques.....                           | 133        |
| 5.2.5. Impact of Increasing Hidden Layer in LSTM.....               | 136        |
| 5.2.6. Increasing Number of Neurons and Layers in Linear Layer..... | 139        |
| 5.2.7. Balancing the Dataset.....                                   | 142        |
| 5.2.8. Tuning Pre-Trained Model Parameters .....                    | 145        |
| 5.2.9. Activation Function.....                                     | 160        |
| 5.3. Conclusion.....                                                | 163        |
| <b>Chapter 6: Discussion .....</b>                                  | <b>165</b> |
| 6.1. Interpreting the Results in Our Context .....                  | 166        |
| 6.1.1. Interpreting Preprocessing Results.....                      | 166        |
| 6.1.2. Interpreting Classification Results .....                    | 167        |
| 6.1.3. Impact of Incorporating Additional Dataset.....              | 170        |
| 6.1.4. Impact of Feature Extraction Techniques .....                | 171        |
| 6.1.5. Impact of Increasing Hidden Layer in LSTM.....               | 174        |
| 6.1.6. Impact of Balancing the Dataset .....                        | 176        |
| 6.1.7. Impact of Tuning Pre-Trained Model Parameters.....           | 178        |

|                                             |            |
|---------------------------------------------|------------|
| 6.1.8. Impact of Activation Function .....  | 183        |
| 6.2. Key Findings .....                     | 184        |
| 6.3. Theoretical Contributions.....         | 192        |
| 6.4. Practical Implication .....            | 194        |
| 6.5. Discussion of Challenges Faced.....    | 196        |
| 6.6. Implications for the NLP Field.....    | 196        |
| 6.7. Comparison with Previous Research..... | 197        |
| 6.8. Sensitivity Analysis.....              | 197        |
| 6.9. Addressing Limitations .....           | 197        |
| 6.10. Resource Allocation .....             | 197        |
| 6.11. Error Analysis .....                  | 198        |
| 6.12. Conclusion.....                       | 203        |
| <b>Chapter 7: Conclusion .....</b>          | <b>204</b> |
| 7.1. Main Findings .....                    | 204        |
| 7.2. Contribution of the Research.....      | 206        |
| 7.3. Limitations and Future Work .....      | 207        |
| 7.4. Conclusion.....                        | 210        |
| <b>References .....</b>                     | <b>213</b> |
| <b>Appendices .....</b>                     | <b>236</b> |
| Appendix A .....                            | 236        |
| Appendix B .....                            | 247        |

Appendix C ..... 249

Appendix D ..... 251

## LIST OF TABLES

|                                                                                                         |     |
|---------------------------------------------------------------------------------------------------------|-----|
| Table 1: Research map .....                                                                             | 18  |
| Table 2: Research structure .....                                                                       | 19  |
| Table 3: Confusion Matrix .....                                                                         | 36  |
| Table 4: Performance comparison of offensive language identification methods .....                      | 45  |
| Table 5: Inclusion and exclusion criteria .....                                                         | 48  |
| Table 6: Search criteria summary .....                                                                  | 49  |
| Table 7: Quality assessment questions.....                                                              | 51  |
| Table 8: Quality assessment results .....                                                               | 51  |
| Table 9: Summary of corpora in the existing literature .....                                            | 72  |
| Table 10: Summary of feature extraction studies in the existing literature .....                        | 77  |
| Table 11: Summary of ATC approaches and their corresponding algorithms in the existing literature. .... | 83  |
| Table 12: Comparative analysis between this systematic review and previous reviews .....                | 88  |
| Table 13: Dataset statistics (Mubarak, Hassan & Chowdhury 2022b) .....                                  | 99  |
| Table 14: Dataset statistics (Mubarak et al. 2021) .....                                                | 100 |
| Table 15: Part from emojis dictionary we have created.....                                              | 110 |
| Table 16: Comparison of sample tweets before and after preprocessing.....                               | 112 |
| Table 17: Hardware configuration .....                                                                  | 122 |
| Table 18: Software configuration.....                                                                   | 123 |
| Table 19:Preprocessing techniques performance metric results .....                                      | 126 |
| Table 20: Classifier results.....                                                                       | 128 |
| Table 21: Comparison of predicted and actual classes for test data (sample) .....                       | 130 |

|                                                                                                          |     |
|----------------------------------------------------------------------------------------------------------|-----|
| Table 22:Performance comparison of the proposed model with prior studies.....                            | 131 |
| Table 23: Performance comparison of the proposed model on OSACT, semEval, and<br>combined datasets ..... | 133 |
| Table 24: Performance metrics of different classifiers and feature extraction techniques .....           | 136 |
| Table 25: Impact comparison of different LSTM layers .....                                               | 139 |
| Table 26: Comparison of NN configurations , LSTM, on accuracy and macro F1 score .....                   | 141 |
| Table 27: Performance comparison of balancing techniques in Arabic text classification ....              | 145 |
| Table 28: Comparison of changing train/ test batch size .....                                            | 150 |
| table 29: Experimental results: warm-up ratios and metrics .....                                         | 155 |
| Table 30: Parameters value .....                                                                         | 159 |

## LIST OF FIGURES

|                                                                                     |     |
|-------------------------------------------------------------------------------------|-----|
| Figure 1: The rule of the category AUSTRALIAN DOLLAR .....                          | 24  |
| Figure 2: Machine learning techniques. ....                                         | 25  |
| Figure 3: Feedforward neural network.....                                           | 27  |
| Figure 4: Convolutional neural network .....                                        | 27  |
| Figure 5: Recurrent neural network .....                                            | 28  |
| Figure 6: Long Short-Term Memory (LSTM) .....                                       | 29  |
| Figure 7: Bert architecture (Devlin et al. 2019) .....                              | 30  |
| Figure 8: Systematic review approach .....                                          | 47  |
| Figure 9: PRISMA flowchart .....                                                    | 50  |
| Figure 10: Progress of publications in terms of domains and publication years. .... | 55  |
| Figure 11: Distribution of publications per country.....                            | 55  |
| Figure 12: ATC domains taxonomy.....                                                | 56  |
| Figure 13: Distribution of ATC studies by domains .....                             | 57  |
| Figure 14: Distribution of ATC topic areas by publication year .....                | 58  |
| Figure 15: ATC tasks and applications across publication years.....                 | 64  |
| Figure 16: Proposed solutions in ATC.....                                           | 86  |
| Figure 17: Proposed framework.....                                                  | 101 |
| Figure 18: Proposed preprocessing submodule.....                                    | 101 |
| Figure 19: Step-by-step process of emoji-to-word interpretation .....               | 106 |
| Figure 20: Proposed emoji-to-word replacement algorithm pseudo code .....           | 110 |
| Figure 21: Proposed architecture for the classification module.....                 | 117 |
| Figure 22: Distribution of sentence lengths in OSACT dataset .....                  | 125 |

|                                                                                                             |     |
|-------------------------------------------------------------------------------------------------------------|-----|
| Figure 23: Performance metrics of various classifiers and feature extraction techniques. ....               | 134 |
| Figure 24: LSTM comparison .....                                                                            | 138 |
| Figure 25: Comparison of NN layers on accuracy and macro F1 score .....                                     | 141 |
| Figure 26: Comparison of balancing techniques for dataset enhancement in Arabic text<br>classification..... | 144 |
| Figure 27: Snapshot of training argument parameters. ....                                                   | 146 |
| Figure 28: Train batch size Vs f1. ....                                                                     | 149 |
| Figure 29: Test batch size Vs acc and f1.....                                                               | 149 |
| Figure 30: Validation loss. ....                                                                            | 154 |
| Figure 31: Custom model implementation with activation functions .....                                      | 161 |

## LIST OF ABBREVIATIONS

|         |                                                            |
|---------|------------------------------------------------------------|
| NLP     | NATURAL LANGUAGE PROCESSING                                |
| ANLP    | ARABIC NATURAL LANGUAGE PROCESSING                         |
| SVM     | SUPPORT VECTOR MACHINE                                     |
| MSA     | MODERN STANDARD ARABIC                                     |
| DA      | DIALECT ARABIC                                             |
| CA      | CLASSICAL (TRADITIONAL OR QURANIC) ARABIC                  |
| KNN     | K-NEAREST NEIGHBORS                                        |
| NB      | NAÏVE BAYES                                                |
| TC      | TEXT CLASSIFICATION                                        |
| ATC     | ARABIC TEXT CLASSIFICATION                                 |
| ML      | MACHINE LEARNING                                           |
| SMOTEEN | SMOTE WITH EDITED NEAREST NEIGHBORS                        |
| BERT    | BIDIRECTIONAL ENCODER REPRESENTATIONS<br>FROM TRANSFORMERS |
| CB      | CLASS-BALANCED                                             |
| RUS     | RANDOM UNDER-SAMPLING                                      |
| ROS     | RANDOM OVER-SAMPLING                                       |

## LIST OF PUBLICATIONS

- A Systematic Review Of Arabic Text Classification: Areas, Applications, And Future Directions. A Wahdan, M Al-Emran, K Shaalan - Soft Computing, 2023
- A Systematic Review Of Text Classification Research Based On Deep Learning Models In Arabic Language. A Wahdan, S Hantoobi, Sa Salloum, K Shaalan - Int. J. Electr. Comput. Eng, 2020
- Text Classification Of Arabic Text: Deep Learning In Anlp. A Wahdan, Sa Salloum, K Shaalan - ... Technologies and Applications: Proceedings Of Amlta ..., 2021
- Enhancing Offensive Arabic Tweet Classification Using Arabert with Fully Connected Nn And Lstm (Under Review)

# Chapter 1: Introduction

## 1.1. Background

In recent years, the widespread adoption of social media platforms has led to an unprecedented amount of user-generated content. While this has facilitated global connectivity and information sharing, it has also given rise to the challenge of offensive language detection and moderation. Offensive language, including hate speech, cyberbullying, and harassment, poses a serious threat to online communities and can have detrimental effects on individuals and society as a whole.(Santos 2022)

The Arabic language presents several modeling challenges in the field of natural language processing (NLP), including its morphological complexity, orthographic ambiguity, variations across dialects, presence of orthographic noise, and limited availability of linguistic resources. It is worth noting that the issues related to right-to-left Arabic typography have largely been addressed and resolved, although implementation may vary (Zahidi, El Younoussi & Azroumahli 2019). In addition, Arabic, a Semitic language predominantly spoken in Arab nations, encompasses three variants: Dialect Arabic (DA), Modern Standard Arabic (MSA), and Classical (Traditional or Quranic) Arabic (CA). These variations exhibit disparities in morphology, syntax, and lexical composition. Low-resourced languages, including Arabic, require greater focus and attention from the NLP community (Al-ghamdi, Al-khalifa & Al-Salman 2023).

Arabic NLP applications encompass various domains, including machine translation, information retrieval, question answering, dialogue systems, and content moderation on social media platforms. Content moderation involves the enforcement of policies to identify and

prevent the posting of user comments containing offensive language, hate speech, cyberbullying, spam, and other inappropriate or harmful content. Detecting such content is challenging due to the sheer volume of posts, dialectal variations, creative spelling used on social media, and the limited availability of data and detection tools, making this area relatively new in the context of Arabic language processing. (Darwish et al. 2021).

The field of NLP (Natural Language Processing) has seen significant advancements in various areas, including the detection of offensive language. Researchers have been actively exploring methods to identify offensive content, which encompasses targeted attacks, vulgar and pornographic language, and hate speech. Initial studies in this field have focused on analyzing comments from news sites, a limited number of tweets, and YouTube comments. Some researchers have concentrated on detecting adult content, while others have specifically aimed at identifying hate speech. Notably, recent benchmarking and shared tasks have specifically targeted the automatic detection of offensive language on platforms like Twitter (Darwish et al. 2021).

Many learning systems up until recently processed natural language using gated RNNs made for processing sequential input, including the LSTM. These designs introduced attention layers to their architecture after introducing the attention mechanism in order to improve learning about the relationships between the sequence's components and weigh the components in accordance with their significance to the model (D. Bahdanau, K.H. Cho 2015)

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that is specifically designed to address the vanishing gradient problem, which is a challenge in training traditional RNNs. LSTM networks can capture and retaining long-term

dependencies in sequential data by utilizing memory cells and various gating mechanisms. The key difference between LSTM and other RNN architectures lies in the structure of the memory cell and the gating mechanisms it employs. The LSTM cell consists of a memory cell that maintains its state over time and three types of gates: the input gate, forget gate, and output gate. These gates control the flow of information into and out of the memory cell. By incorporating these gating mechanisms and memory cells, LSTM networks can effectively capture and retain long-range dependencies in sequential data. This makes them well-suited for tasks that involve analyzing and generating sequences, such as natural language processing and speech recognition (Hochreiter & Schmidhuber 1997a). The ability to handle long-term dependencies sets LSTM apart from other RNN architectures and makes it particularly useful in scenarios where context over longer sequences is crucial. (Strubell 2019).

Transformers have mostly supplanted the prior models in sequence in recent years due to their superior performance in NLP tasks. Transformers are solely based on the attention mechanism due to the fact that this mechanism is capable of achieving the RNN's expected learning performance (Vaswani et al. 2017). The BERT is one of the newest and most promising transformer models. (Farokhian, Rafe & Veisi 2022). In recent years, the development of deep pre-trained models such as GPT and BERT has gained traction in the field of NLP. These models utilize large-scale unlabeled corpora and leverage self-supervised learning techniques. By harnessing the semantic meanings of words and employing advanced architectures like Transformers, these models can capture significant linguistic knowledge from text (Al-ghamdi, Al-khalifa & Al-Salman 2023). This progress in pre-trained models and their ability to understand language has contributed to the advancement of offensive language detection research. These models can be fine-tuned and utilized to build efficient and effective

systems for detecting offensive content in various contexts, including social media platforms like Twitter.

BERT, a bidirectional deep Transformer, has had a significant impact on the advancement of pre-trained models. The process of adapting BERT for specific NLP tasks involves two primary ways: feature extraction and fine-tuning. In feature extraction, the original architecture of the BERT model remains unchanged, and its parameters are "frozen". The pre-trained BERT model is used to extract features, which are then passed into a separate classifier model to address a specific task. On the other hand, in fine-tuning, the parameters of the BERT model are adjusted by introducing additional layers to the existing architecture. These new layers enable the model to be trained on specific downstream tasks (Devlin et al. 2019).

Pretrained models, such as BERT, have been trained on vast amounts of data and have learned rich representations of language. These models capture intricate patterns and semantic relationships in text, which can be leveraged for various downstream tasks, including text classification, named entity recognition, sentiment analysis, and more. By using a pretrained model as a starting point, we can benefit from the knowledge it has acquired and save significant training time and resources. (Devlin et al. 2019).

Multiple BERT models have been pre-trained to cater to the Arabic language such as multilingual BERT, AraBERT, MARBERT, ArabicBERT, XLM-RoBERTa, QARiB, GigaBERT, Arabic ALBERT are introduced to achieve similar success as BERT did for English (Alammary 2022). For instance, Devlin and colleagues introduced a multilingual model capable of supporting over 100 languages, including Arabic. Similarly, Abdul-Mageed

and co-authors utilized a dataset of 1 billion tweets to train a specialized Arabic BERT model known as MARBERT (Abdul-Mageed, Elmadany & Nagoudi 2021). Additionally, Antoun and collaborators developed an Arabic-specific model named Arabert, which underwent pre-training using approximately 24 gigabytes of text (Antoun, Baly & Hajj 2020). However, AraBERT outperforms multilingual BERT and other approaches in various Arabic NLP tasks, demonstrating state-of-the-art performance (Antoun, Baly & Hajj 2020).

AraBERT, a language model for the Arabic language, is based on Google's BERT architecture and follows the same BERT-Base configuration introduced by (Antoun, Baly & Hajj 2020). To train this model, two publicly available large Arabic corpora were utilized. The first corpus, called the 1.5 billion words Arabic Corpus, comprises over 5 million articles sourced from ten major news outlets across eight countries. The second corpus, known as OSIAN (Open-Source International Arabic News Corpus), consists of approximately 3.5 million articles (equivalent to about 1 billion tokens) obtained from 31 news sources in 24 Arab countries. After removing duplicate sentences, the final pre-training dataset comprised 70 million sentences, totaling around 24GB of text. This diverse dataset represents news from various media outlets across different Arab regions, encompassing a wide range of topics discussed in the Arab world. Notably, words containing Latin characters were retained to preserve named entities, scientific or technical terms mentioned in their original language, ensuring valuable information is not lost (Antoun, Baly & Hajj 2020). Following pre-training, the model underwent fine-tuning on three tasks: Sequence Classification, Named Entity Recognition, and Question Answering.

AraBERT has shown great promise in various NLP tasks, and we hypothesize that it can significantly improve the performance of Arabic offensive text classification. The study conducted by Abu-farha & Magdy (2020) demonstrated that even after fine-tuning, AraBERT outperformed other models, achieving state-of-the-art performance on Arabic dialects. Therefore, AraBERT was chosen as the base for the classification module. Additionally, Alzu'bi et al. (2022) utilized AraBERT, which achieved the highest accuracy among other pre trained models.

One of the most utilized approaches for text classification tasks combines the knowledge embedded in the pre-trained deep bidirectional transformer BERT (Devlin et al. 2019) with Convolutional Neural Networks (CNN) for text (Kim 2014). This combination of models has been shown to yield better results than using BERT or CNN individually (Safaya, Abdullatif & Yuret 2020). On the other hand, Long Short-Term Memory (LSTM) has shown great promise. It has consistently demonstrated high efficiency in text classification studies, as identified in a previous systematic review conducted in 2020 (Wahdan et al. 2020b).

On the other hand, combining pretrained models with neural networks offers several advantages. Neural networks can fine-tune pretrained models on specific tasks. Fine-tuning involves updating the parameters of the pretrained model using task-specific labeled data. This allows the model to adapt its learned representations to the specific nuances and characteristics of the target task. Fine-tuning helps the model generalize better and improves its performance on the specific task at hand. Overall, combining neural networks with pretrained models provides a powerful framework for leveraging pre-existing linguistic knowledge, fine-tuning on specific tasks, and facilitating transfer learning. This approach offers improved

performance, efficiency, and scalability in various natural language processing tasks (Farokhian, Rafe & Veisi 2022).

In this work, we aim to enhance the accuracy of Arabic offensive tweet classification by leveraging state-of-the-art techniques and models. Specifically, we explore the effectiveness of the AraBERT model a pre-trained language model specifically designed for Arabic text processing in conjunction with fully connected neural networks (NN) and long short-term memory (LSTM) networks.

However, AraBERT outperforms multilingual BERT and other approaches in various Arabic NLP tasks, demonstrating state-of-the-art performance (Antoun, Baly & Hajj 2020). This motivated us to introduce an ensemble methodology for classifying Arabic offensive Arabic tweet to achieve better results by combining the most recent pretrained model with deep learning technology. Specifically, we explore the impact of preprocessing techniques, such as AraBERT preprocessing, emoji-to-word interpretation, and punctuation removal, on the text classification accuracy. The OSACT dataset was chosen since it has already been used in studies on abusive text classification on Twitter. Because of its coverage of offensive language categories and extensive annotation, the dataset is useful for assessing and comparing Arabic text classification models. Experiments were also carried out on the semEval dataset with the suggested model.

For our base model, we utilize AraBERT v2 (Antoun, Baly & Hajj, 2020), which is publicly available in the Hugging Face library. In our proposed model, we enhance AraBERT by incorporating an LSTM layer and two linear layers., in conjunction with fully connected neural networks (NN) and long short-term memory (LSTM) networks.

To evaluate the performance of our proposed approach, we utilize multiple metrics, we can assess the generalizability of our model across different variants of the Arabic language.

Our work builds upon existing research in Arabic offensive text classification and extends it by incorporating advanced models and techniques. By focusing on less-resourced languages, we contribute to the broader goal of creating inclusive NLP frameworks that cater to the linguistic diversity of online communities. Moreover, our findings will provide valuable insights and practical solutions for addressing offensive language in Arabic social media content.

## **1.2 Motivation**

The widespread adoption of social media platforms in recent years has revolutionized communication and information sharing, connecting people from diverse backgrounds across the globe. However, this surge in user-generated content has also brought about significant challenges, particularly in the form of offensive language and its impact on online communities. Offensive language, including hate speech, cyberbullying, and harassment, poses a grave threat to individuals and society as a whole, compromising the safety, well-being, and inclusivity of online spaces.(Mubarak, Hassan & Chowdhury 2022a)

Arabic language is one of the most important natural languages worldwide, where it occupies the 4th rank. Arabic speakers have reached more than 300 million in population. Its usage is widespread worldwide, and it is the backbone of the Arab world. Since Arabic is the language of the Qur'an, some Islamic countries adopt it as their primary language. Arabic, a Semitic language, is distinguished by its complex morphology and sophisticated syntax (Al-Emran, Zaza & Shaalan 2015). There are 28 letters in the Arabic alphabet, including hamzas

and elmodod. Each letter has different structures depending on where it is located in the word: at the start, middle, or end (Wahdan et al. 2020b).

Arabic language processing is challenging due to its ambiguity and richness of morphology and structure (Elayeb 2019; Zahidi, El Younoussi & Azroumahli 2019). The continuous efforts spent by scholars in the domain of Arabic natural language processing (ANLP) have provided solutions to various problems and challenges posed by the language and led to rapid improvements in essential tools and linguistic resources (Hassan et al. 2020). ANLP has gained more attention from the research community. This is witnessed in the significant growth in research, applications, and systems. However, ANLP is still underdeveloped compared to other NLP systems in languages other than Arabic (Zahidi, El Younoussi & Azroumahli 2019). Nowadays, text classification or text categorization has become a crucial topic due to the Internet revolution that has amplified the massive amount of available text. The large amount of data accessible from various sources, including texts shared through social media, Internet sites, news pages, e-mail messages, reports, and journal articles, has dramatically enhanced research on text classification (Elhassan & Ahmed 2015). This unmanageable amount of data is on the rise. Text classification is the process of automatically categorizing a series of documents into distinct categories based on the content of the text itself, using technologies and algorithms (Mohammad & Al-momani 2016). The importance of text classification reflects the number of applications that appeared, where text classification is applied in different areas using various methodologies and approaches

In the realm of Arabic NLP, addressing the issue of offensive language detection and moderation is of utmost importance. The Arabic language presents unique modeling

challenges, such as its morphological complexity, orthographic ambiguity, dialectal variations, and the presence of orthographic noise. Furthermore, the scarcity of linguistic resources dedicated to Arabic and the limited availability of detection tools further compound the difficulties faced in effectively identifying and mitigating offensive language in Arabic text. Given the linguistic richness and diversity of Arabic, it is essential to develop robust and accurate methods for detecting and addressing offensive language to ensure the safety and well-being of Arabic-speaking online communities (Darwish et al. 2021).

Despite previous research efforts in offensive text classification, there are still significant gaps and limitations that need to be addressed in the context of Arabic NLP. Existing studies have often focused on specific aspects of offensive language or utilized limited datasets, thereby failing to provide comprehensive solutions that can effectively handle the wide range of offensive language expressions present in Arabic text. This research gap underscores the need for further investigation and improvement in offensive language detection and moderation techniques specific to the Arabic language.(Almaliki et al. 2023)

The practical implications of developing accurate and efficient methods for offensive language detection in Arabic are substantial. Effective content moderation can foster healthy online interactions, protect vulnerable individuals from harm, and promote inclusivity within Arabic-speaking online communities(Shannag, Hammo & Faris 2022). By addressing the challenges unique to Arabic NLP and advancing the state of offensive text classification in Arabic, our research aims to contribute to the creation of safer and more inclusive online environments, aligning with the broader societal goal of combating online abuse and ensuring a positive digital experience for all users.

Additionally, from an academic standpoint, this research holds significant value. By integrating state-of-the-art techniques, leveraging pre-trained language models, and exploring the effectiveness of specific neural network architectures, we aim to advance the field of Arabic NLP and contribute to the existing body of knowledge. Our study seeks to improve upon existing methodologies and algorithms for offensive language detection and provide valuable insights into effective approaches for handling offensive language in Arabic text, thereby addressing a critical research gap and expanding the frontiers of Arabic NLP.

In conclusion, our research is motivated by the pressing need to tackle offensive language in Arabic text and its adverse effects on online communities. By developing robust and accurate methods for offensive language detection and moderation in Arabic, we aim to enhance the safety, inclusivity, and well-being of Arabic-speaking online platforms. Furthermore, our work contributes to the broader field of Arabic NLP, advancing the state of the art and opening avenues for further research in cross-cultural and multilingual offensive language detection.

### **1.3 Problem Statement**

The detection and moderation of offensive language in Arabic text pose significant challenges that need to be addressed in the field of NLP. Existing methodologies and algorithms for offensive text classification often fall short in accurately and comprehensively identifying offensive language expressions in Arabic due to several key factors.

Firstly, the Arabic language exhibits complex morphological structures, including rich inflectional and derivational processes, making it difficult to capture the various forms and variants of offensive language. Furthermore, Arabic text is characterized by orthographic

ambiguity, where multiple words or phrases can be represented by the same sequence of characters, leading to misinterpretation and inaccurate classification of offensive content (Darwish et al. 2021).

Secondly, the presence of dialectal variations in Arabic further complicates the task of offensive language detection. Different dialects may employ unique linguistic expressions, slang, or cultural references that are specific to regions or communities. This dialectal diversity necessitates the development of tailored models and approaches to effectively capture and classify offensive language in different dialects (Touahri, I., & Mazroui 2021).

Additionally, the prevalence of orthographic noise in Arabic text, such as misspellings, abbreviations, and non-standard writing styles commonly found in social media platforms, further challenges the accurate identification of offensive content. Creative spelling variations and unconventional word formations require robust techniques that can handle such noise effectively (Al-Jarf 2023).

Moreover, the scarcity of linguistic resources dedicated to Arabic offensive language detection and the limited availability of annotated datasets hinder the development and evaluation of accurate models. The lack of sufficient training data specific to offensive language in Arabic limits the performance and generalizability of existing approaches, demanding the exploration of alternative strategies for addressing this scarcity.(Awal, M. R., Lee, R. K. W., Tanwar, E., Garg, T., & Chakraborty 2023)

Furthermore, the existing research on offensive language detection in Arabic has primarily focused on specific subsets of offensive language, such as hate speech or adult content, while neglecting the broader spectrum of offensive language expressions. This

limitation in coverage hampers the overall effectiveness and practical applicability of existing models in real-world scenarios where offensive language can take diverse forms (Ferrag, M. A., Friha, O., Kantarci, B., Tihanyi, N., Cordeiro, L., Debbah, M., ... & Choo 2023).

Additionally, the absence of a standardized Arabic benchmark dataset for evaluating and comparing Arabic Text Classification (ATC) models, coupled with the limited exploration of ensemble models that combine pretrained models with neural networks, underscores the existing research gap in this domain. This gap has been highlighted in both the Systematic Review of this thesis and in the works of Wahdan et al. (2020). Only Nessir et al. (2022) explored the combination of pretrained models with neural networks among the studies.

The study addresses several key problems in the field of Arabic offensive text classification. Firstly, existing approaches have proven to be ineffective in accurately identifying offensive language. To overcome this limitation, the research aims to develop an innovative framework that leverages the combination of pretrained models and neural networks in ensemble models. This framework is designed to significantly enhance the accuracy of offensive text classification in Arabic, providing a more robust solution.

Secondly, there exists a noticeable gap in the domain of offensive language detection and moderation in Arabic. To address this gap, the study focuses on developing methods that are both reliable and accurate in detecting offensive language in Arabic text. By exploring state-of-the-art techniques, including pre-trained language models and advanced neural network architectures, the research aims to raise the standards of offensive language detection and moderation.

Furthermore, the scarcity of labeled datasets poses a significant challenge in training effective models for offensive language classification in Arabic. To tackle this issue, the study investigates the potential of transfer learning techniques. By utilizing transfer learning, the model can benefit from knowledge gained in related domains or from other languages, enhancing its performance and generalizability despite limited labeled data.

Overall, the research endeavors to provide comprehensive solutions to these problems, ultimately contributing to the advancement of offensive language detection and moderation in Arabic text. The proposed framework, along with the novel techniques explored, aims to significantly improve the accuracy, coverage, and efficiency of offensive language classification, making it an asset for decision makers, developers, and policymakers across various domains.

## **1.4 Research Questions**

RQ1: What are the research gaps in ATC?

RQ2: What are the effects of utilizing pre-processing methods on the performance of Arabic Text Classification?

RQ3: What is the most effective model for improving the accuracy of Arabic offensive text classification using pre-trained models?

RQ4: To what extent does the proposed model be able to classify Arabic offensive text?

## 1.5 Research Aim and Objectives

By addressing these critical limitations just mentioned in section 1.3 and developing effective solutions, this research aims to develop robust and accurate methods for offensive language detection and moderation in Arabic text by leveraging state-of-the-art techniques, including pre-trained language models and advanced neural network architectures. The goal is to improve the accuracy, coverage, and efficiency of offensive language classification, overcome the scarcity of labeled datasets, and address the specific challenges of the Arabic language in online contexts. The primary objectives of this study are as follows:

- To identify the research gaps in Arabic text classification.
- To identify the effects of utilizing pre-processing methods on the performance of Arabic Text Classification.
- To develop the most effective model for improving the accuracy of Arabic offensive text classification using pre-trained models.
- To validate the proposed model be able to classify Arabic offensive text.

By addressing these critical limitations and developing effective solutions, this research aims to contribute to the advancement of offensive language detection and moderation in Arabic NLP. The proposed methods and findings will not only enhance the safety and well-being of Arabic-speaking online communities but also pave the way for further research in the broader field of multilingual offensive language detection and moderation, fostering inclusive and secure online environments across different languages and cultures.

## 1.6 Research Significance

The significance of this research lies in its potential to address the pressing issue of offensive language detection and moderation in Arabic-speaking online communities. With the rapid growth of social media and online platforms, these communities have become increasingly vulnerable to the harmful effects of offensive and abusive content. However, the existing research and resources in this domain are predominantly focused on English or other widely spoken languages, leaving a significant gap in effective solutions for Arabic offensive language detection and moderation.

By conducting this research, we aim to contribute to the development of tailored and effective approaches for detecting and moderating offensive language in Arabic. The findings of this study will provide valuable insights into the unique characteristics and challenges associated with Arabic offensive text, paving the way for more accurate and robust detection models and moderation systems. This, in turn, will enable online platforms and community moderators to create safer and more inclusive environments for Arabic-speaking users, fostering healthy and respectful online interactions.

Moreover, the practical implications of this research extend beyond the realm of online communities. The insights gained from this study can be leveraged by various stakeholders, including social media platforms, content moderators, and policymakers, to design and implement proactive strategies for combating offensive language in Arabic. By integrating the findings into their moderation processes and policies, these stakeholders can work towards mitigating the harmful impact of offensive content, protecting users from online harassment, and promoting a positive online culture.

Ultimately, the significance of this research lies in its potential to contribute to the well-being and safety of Arabic-speaking online communities. By addressing the specific challenges of offensive language detection and moderation in Arabic, this study strives to create a positive ripple effect, fostering a more inclusive and respectful online environment for Arabic speakers and promoting cross-cultural understanding.

Table 1 provides a concise research map that outlines the core elements of our study on the accurate detection and moderation of offensive language in Arabic text. It serves as a navigational guide for understanding the scope and objectives of our research endeavor. At its heart, our research aims to develop robust methods for effectively identifying and moderating offensive language in Arabic text, while addressing the unique challenges posed by the language's intricate morphology, dialectal variations, orthographic complexity, and the scarcity of labeled datasets. To achieve this, we have formulated specific research questions (RQ1 to RQ4) that guide our investigation. These questions delve into identifying research gaps in Arabic text classification, exploring the impact of pre-processing methods, determining the most effective model for enhancing accuracy, and validating the proposed model's capability to classify offensive text effectively. In essence, Table 1 encapsulates the roadmap that charts our course towards enhancing the quality and efficiency of offensive language detection in the Arabic language, within the context of online communication.

**Table 1: Research map**

| Research Problem                                                                                                                                                                                                                                                                                                             | Research Aim                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Research Questions                                                                                                                 | Research objectives                                                                                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| The accurate detection and moderation of offensive language in Arabic text, considering the challenges posed by the language's complex morphology, dialectal variations, orthographic ambiguity, orthographic noise, limited linguistic resources, and the need for comprehensive coverage of offensive language expressions | To develop robust and accurate methods for offensive language detection and moderation in Arabic text by leveraging state-of-the-art techniques, including pre-trained language models and advanced neural network architectures. The goal is to improve the accuracy, coverage, and efficiency of offensive language classification, overcome the scarcity of labeled datasets, and address the specific challenges of the Arabic language in online contexts | RQ1: What are the research gaps in ATC?                                                                                            | To identify the research gaps in Arabic text classification                                                                     |
|                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RQ2: What are the effects of utilizing pre-processing methods on the performance of Arabic Text Classification?                    | To identify the effects of utilizing pre-processing methods on the performance of Arabic Text Classification                    |
|                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RQ3: What is the most effective model for improving the accuracy of Arabic offensive text classification using pre-trained models? | To develop the most effective model for improving the accuracy of Arabic offensive text classification using pre-trained models |
|                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                | RQ4: To what extent does the proposed model be able to classify Arabic offensive text?                                             | To validate the proposed model be able to classify Arabic offensive text                                                        |

## 1.7 Research Structure

Table 2 shows the research structure.

**Table 2: Research structure**

| Chapter #        | Summary                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Chapter 1</b> | This chapter serves as the introduction, summarizing the study, defining the problem, and outlining the study's objectives. It provides context, motivation, and research questions while explaining the methods employed.                                                                                                                                                                                                                                                                      |
| <b>Chapter 2</b> | This chapter offers an overview of text classification techniques, encompassing manual and statistical methods, machine learning techniques, and neural networks, transformers. Also, it discusses the various metrics used to assess the performance of the proposed approach or model in Arabic text classification                                                                                                                                                                           |
| <b>Chapter 3</b> | In this chapter, a comprehensive systematic review of existing literature on Arabic text classification is presented. It also explores the application of systematic review techniques in Arabic text classification, focusing on related works related to the classification of offensive tweets.                                                                                                                                                                                              |
| <b>Chapter 4</b> | The methodology chapter describes the study's approach, emphasizing two key components: the framework and dataset description. It provides a detailed explanation of the proposed model.                                                                                                                                                                                                                                                                                                        |
| <b>Chapter 5</b> | This chapter present the experimental setting, and the experimental results of the study. Also, it will compare the Research results with the previous related work.<br>This chapter focuses on the evaluation of performance metrics in the context of the study. Furthermore, the chapter presents the experimental results obtained from the conducted study and conducts a comparative analysis with previous related work to establish the novelty and contribution of the research study. |
| <b>Chapter 6</b> | It discusses the results obtained in Chapter 5. This chapter presents the key findings of the study, along with their theoretical and practical implications. Additionally, it conducts an error analysis to further explore these implications.                                                                                                                                                                                                                                                |
| <b>Chapter 7</b> | The final chapter outlines the research's limitations, draws conclusions, and offers recommendations for future research and work in the field.                                                                                                                                                                                                                                                                                                                                                 |

## Chapter 2: Technical Background

In the Technical Background chapter, we will embark on an in-depth exploration of various techniques used in text analysis for both feature extraction, classification, transformers, and performance evaluation. Firstly, we will discuss feature extraction methods such as Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), Glove, and Word2Vec.

Following our discussion on feature extraction, we will then shift our focus to classification techniques. These will encompass statistical methods, machine learning techniques, and neural networks, including Feedforward Neural Networks (FNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM). Additionally, we will touch upon the concept of transformers and their pivotal role in text classification.

Moreover, we'll dive deep into the evaluation of performance metrics in the context of the study. It discusses the various metrics used to assess the performance of the proposed approach or model in Arabic text classification. These metrics include accuracy, precision, recall, F1-score, and the concept of the confusion matrix, which provide quantitative measures of the classification system's effectiveness. These metrics are the compass by which we gauge the performance of our text analysis techniques, ensuring that our insights are both reliable and actionable.

## 2.1. Feature Extraction

The process of converting a text document into a numerical vector is known as feature extraction or feature representation. Various methods are employed for feature extraction, including Bag of Words (BoW) and TF-IDF (Term Frequency-Inverse Document Frequency) approaches. (Alhaj et al. 2018)

In a Bag of Words (BoW) approach, the arrangement of terms within documents is disregarded, and their frequencies of occurrence are utilized (El-Alami et al. 2020). Consequently, each unique word in a text collection is treated as a distinct feature. This results in a multi-dimensional feature vector representing each document. In this feature vector, each dimension is associated with a value that is determined by both Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) (Alhaj et al. 2018).

Specifically, for a collection of documents  $D$  containing  $N$  documents, where  $D = \{d_1, d_2, \dots, d_n\}$ , each document  $d_n$ , which consists of a set of terms, is represented as vectors in the Vector Space Model (VSM) as shown in equation 1:

$$d_{ij} = (t_{1j}, t_{2j}, \dots, t_{ij}), j = 1, 2, \dots, y, (1)$$

Here,  $y$  represents the number of distinct words in the document  $d_i$ .

The TF-IDF technique employs TF and DF to calculate the weight of a word in a document using Equations (2) and (3):

$$IDF(t) = \log(N) / DF(d, t) (2)$$

$$TF-IDF = TF(t, d) * IDF(d, t)$$

Where  $DF(d, t)$  is the count of documents containing term  $t$ ,  $N$  is the total number of documents in the training set, and  $TF(t, d)$  is the frequency of term  $t$  occurring in document  $d$ .

The traditional approach to text modeling, which relies on the Bag-Of-Words (BOW) representation, used to achieve top-notch results in the field of Natural Language Processing (NLP). However, it had limitations, such as suffering from the curse of high dimensionality and lacking the ability to capture semantic relationships between different text elements. The introduction of pre-trained word embedding models like Word2vec (Kim 2014), Doc2vec (El-Halees 2017), and Glove (Undavia, Meyers & Ortega 2018) brought about a new set of effective text representation models for tasks like text classification and other NLP applications. These pre-trained models, trained on extensive text corpora, aimed to create continuous distributed representations for text, representing words as dense vectors in lower-dimensional vector spaces. While word embedding demonstrated good performance, it didn't account for relationships among multiple words and the complete meaning of sentences within the text.

To address these challenges, contextual embedding models have proven to be effective in learning universal sentence representations. These pre-trained models allow for the consideration of word context and sentence meaning. For instance, Embeddings from Language Models, known as ELMo (Strubell 2019) is a dynamic language modeling technique that learns embeddings based on the context, even for polysemous words. It combines two language models in both directions, forming a multi-layer Recurrent Neural Network (RNN). ULMFiT (Howard & Ruder 2018) on the other hand, is a left-to-right language model that incorporates various fine-tuning techniques to enhance the performance

of pre-trained models like ELMo. In contrast to ELMo, which concatenates a left-to-right and right-to-left model, BERT (Devlin et al. 2019). employs bidirectional transformer training to generate more accurate word representations. Several versions of BERT exist that support the Arabic language, including the multilingual BERT (mBERT) and two AraBERT variants.

## **2.2. Classification Techniques**

There are two basic methods of classifying texts. One is the grammar or linguistic based method, which depends on preparing certain rules and operating them in some expert systems to classify the texts. The second method is the process of feeding the text extrapolation process with a set of documents called training documents that have previously been classified according to specific categories (Salloum et al. 2017). Below are the explanations of the two methods.

### **2.2.1. Manual and Statistical Techniques**

Manual text classification (TC) involves the creation of specific queries for different categories, such as sports, nutrition, clothing, and health. These queries are then used within a search engine to categorize the entered text based on predefined criteria. While effective for small texts, this method is impractical for large texts or numerous documents. The accuracy of manual classification relies on the validity of queries and the expertise in designing them. Expert systems, like Construe-TIS designed for Reuters News, employ rules and inquiries to automate the classification process. For instance, Construe-TIS successfully applied 674 categories and identified over 170,000 companies, utilizing a classification approach based on concepts rather than individual words, as illustrated in Figure 1 (Cohen 2004).

```

\ **
test:
  (or [australian-dollar-concept]
    and [dollar-concept]
      [australia-concept]
      (not [us-dollar-concept])
      (not [singapore-dollar-concept]))

```

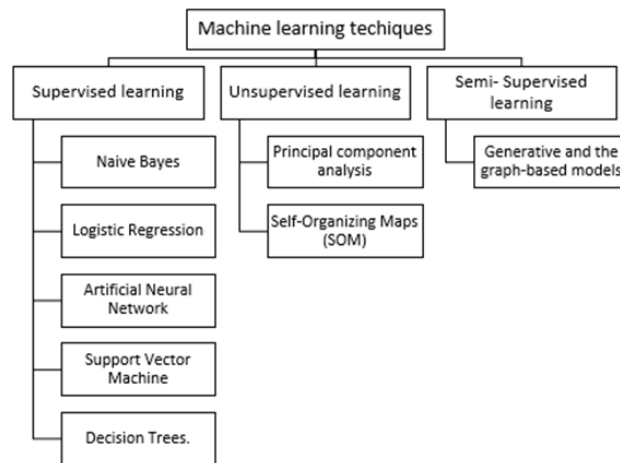
**Figure 1: The rule of the category AUSTRALIAN DOLLAR**

Statistical text classification refers to techniques grounded in mathematical rules and principles. These methods, including Frequentist procedures, Bayesian procedures, and Binary and multiclass procedures, are well-suited for handling small datasets, as noted by” (Abdeen 2019).

### 2.2.2. Machine Learning Techniques

Due to the large amounts of information during the past two decades, there emerged a need for technical methods of classifying huge texts as statistical and manual methods had never become useless in this field (Abandah 2015; Lee 2017; Al-Smadi 2019; Zhong 2019; S. A. Salloum 2020; Salloum 2020b, 2020a). Therefore, machine learning classification techniques appeared which aim to classify unstructured texts and documents based on certain algorithms designed for this purpose. Machine learning techniques can be divided into groups as shown in Figure 2 (Abdeen 2019). In the Supervised Learning technique, different data are recognized, whether visual, audio, or text. The data are compared with the expected ideal data or results through a process called backpropagation in which the data are directed from the output layers to the input layers and errors are corrected and minimized to achieve better

accuracy. In Unsupervised Learning, the learning process takes place during the processing. There are no previous data for comparison as the network analyzes the data and processes them; then, builds a function to determine the error rate and reduce it to obtain high accuracy. The Semi-Supervised learning technique is a combination of the previous two techniques. Researchers saw that there was little unnamed data that could contribute to improving the accuracy of the learning process that took place during the data classification stage. So, a small percentage of the data was used in Semi-Supervised learning.



**Figure 2: Machine learning techniques.**

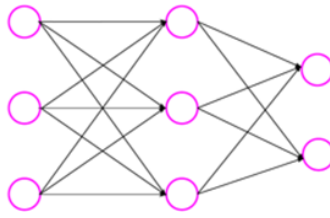
### 2.2.3. Neural Network

Algorithms of neural networks give very accurate results in the area of NLP and support deep learning very significantly as they solve some errors and problems related to the variance of data resulting from the process of deep learning (Alhashmi 2020). Neural networks are used in the classification of texts to address linear and non-linear problems. The backpropagation model classifies texts using neural networks by means of a group of nodes that form and represent a mathematical model of biological nerves. No interference and self-

learning affect these neural networks as they are used to identify systems and patterns and to classify text and image processing (Harrag 2009a). Many consider neural networks to be units that are connected to each other and intertwined similarly to the dynasticity of the neuron. In addition, through this process of entanglement, the inputs are made, and the other neuron allows the exit of the transmitted outputs. The design of ANN is similar in structure to that of the brain and its neural network. All types of sabotage of the neural network have a basic principle, which is that every neuron in the network receives inputs, processes them, and sends outputs. Each neuron is linked to at least one neuron, and each connection or association has a specific digital weight called the weight factor, which works to reflect the importance of the communication between neurons in the network (D. Svozil 1997). Some popular primary types of neural networks follow.

#### **2.2.3.1. Feedforward Neural Network (FNN)**

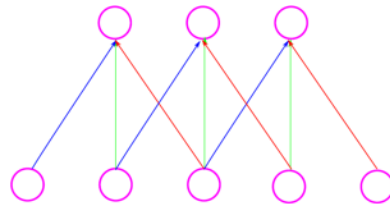
It is a neural network whose cells are linearly connected to each other and do not represent a cycle like any other neural network. The single layer perceptron (SLP) is the simplest form of this type of neural network since the inputs are directly related to the output. In the SLP, the inputs go through several layers of transformation, so they are considered very suitable for categorizing texts (Rumelhart 1986). The structure of the feedforward neural network is shown in Figure 3.



**Figure 3: Feedforward neural network**

#### 2.2.3.2. Convolutional Neural Network (CNN)

Convolutional neural networks are inspired by the Multi-Layer Perceptron (MLP) and are designed to extract the spatial structure in image data and the positions of objects that can be used in the image (Sainath 2013). The same principle and idea were used to classify texts and one-dimensional words, and this was done through interaction with neighboring neurons. Its name is “convolutional” due to folding process that occurs between neurons during the classification process. The structure of the convolutional neural network is shown in Figure 4.

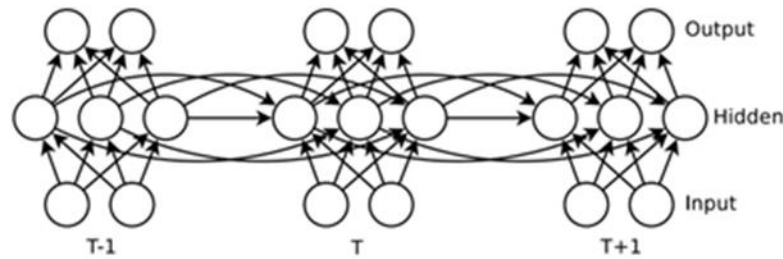


**Figure 4: Convolutional Neural Network**

#### 2.2.3.3. Recurrent Neural Network (RNN)

The structure in this type of network is done continuously and sequentially since the outputs of the previous stage are the inputs of the next stage. By contrast, with traditional neural networks, the inputs and outputs are separate units from each other. Sometimes there is a need to predict the desired word, and, therefore, the need develops to refer to the words prior

to the review, and this is what recurrent neural networks do. A hidden layer has been used to help with this type of network and to complete the process (Sutskever 2011). An example of a frequent neural network is shown in Figure 5.

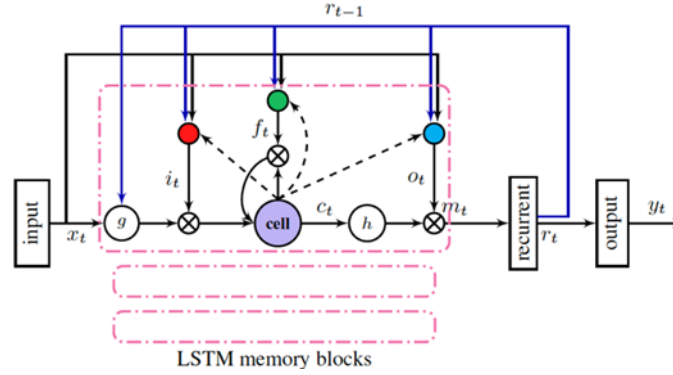


**Figure 5: Recurrent neural network**

#### **2.2.3.4. Long Short-Term Memory (LSTM)**

LSTM are long short-term memory networks as they are a specific recurrent neural network structure (RNN) designed for use in modeling long-range time series that are observed to operate more precisely than RNN. The neurons of this network are made up of units that are gates; in this way, the network can control the flow of the inputs that lead to the final output. Thus, only a few inputs may participate in the output, so the error rate is reduced (Hochreiter & Schmidhuber 1997b). LSTM contains units called memory blocks, and they are in the hidden layer. These blocks have self-connections in memory cells that record the time status of the network during the work. They also contain units called gates that contribute to controlling the flow of inputs and outputs. Their input gates control the activation process for entering information into the memory. As for the output port, it controls the output after the activation process, which takes place at the entry gate. A portal called the forget gate, which

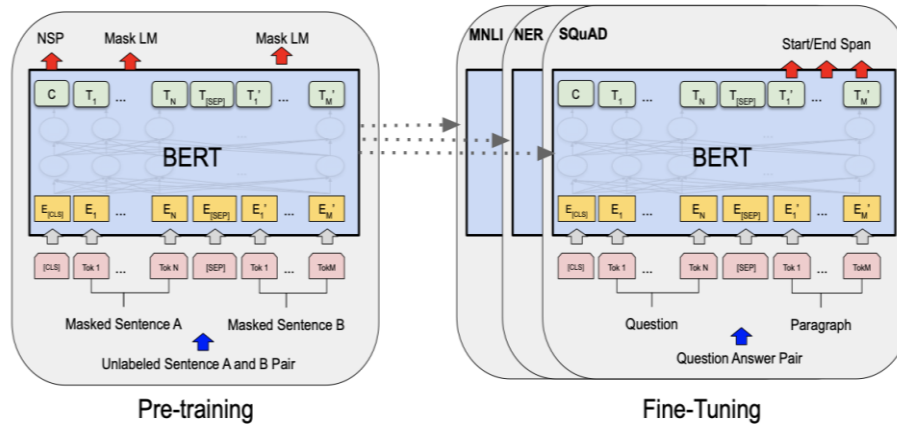
addresses the weakness of LSTM in determining flows for specific units, has also been added (Lee 2017). Figure 6 shows the structure of LSTM RNN.



**Figure 6: Long Short-Term Memory (LSTM)**

## 2.3. Transformers

Bidirectional Encoder Representations from Transformers (BERT) is a language representation model that emerged in 2018, created by Jacob Devlin and his team at Google (Devlin et al. 2019).. Since its inception, it has become a widely adopted baseline in the field of natural language processing. Unlike earlier language models that only considered context in one direction, BERT was designed to be bidirectional, allowing it to predict words based on both preceding and following context (Devlin et al. 2019). Figure 7 explains the BERT model, a multi-layer bidirectional Transformer encoder. It incorporates parameters such as the number of layers (L), hidden size (H), and self-attention heads (A). This work primarily focuses on two model sizes: BERTBASE (L=12, H=768, A=12) and BERTLARGE (L=24, H=1024, A=16). Unlike Open AI GPT, BERTBASE employs bidirectional self-attention. The input representations are designed to accommodate both single sentences and pairs of sentences in a unified token sequence. The use of Word Piece embeddings and a special classification token ([CLS]) at the sequence's beginning facilitates downstream tasks. (Devlin et al. 2019)



**Figure 7: Bert architecture (Devlin et al. 2019)**

Additionally, BERT was developed as an unsupervised model, capable of training on vast amounts of plain text data readily available on the internet in numerous languages. This combination of features has enabled BERT to excel in various natural language processing tasks, including text classification (Alammary 2022). There are primarily two approaches to utilizing BERT: feature extraction and fine-tuning (Alammary 2022). In feature extraction, the original BERT model architecture remains intact, with its parameters fixed. Features are extracted from the pre-trained BERT model and subsequently input into a separate classifier model to address a specific task. In contrast, fine-tuning involves adjusting the model's parameters by adding additional layers to the original BERT structure. These added layers facilitate training the model on specific downstream tasks (Alammary 2022).

In the initial paper authored by Devlin, Chang, Lee, and Toutanova (Devlin et al. 2019), two BERT models were introduced: BERT-large and BERT-base, both of which were designed for English. They were pre-trained on extensive internet-derived corpora, making them computationally demanding with substantial memory requirements. As BERT garnered

increased attention from researchers and practitioners, numerous variations of BERT models were developed. These newer models aimed to address limitations in the original versions, either by enhancing overall performance or by optimizing inference speed (Alammary 2022). Additionally, there were models tailored to support languages beyond English (Alammary 2022).

### **2.3.1. Arabic Language Models**

Several language models have been developed for the Arabic language, with one of the most prominent being AraBERT (Antoun, Baly & Hajj 2020). AraBERT shares its foundation with the BERT architecture (Devlin et al. 2019), and adopts the same BERT Base configuration. In its second iteration, AraBERTv1, the Farasa tool is utilized for text segmentation prior to training. The training data for AraBERT comprises a substantial 23GB of Arabic text, encompassing over 70 million sentences and a staggering 3 billion words (Antoun, Baly & Hajj 2020). constructed this training corpus by drawing from Arabic Wikipedia, the Open Source International dataset (OSIAN) (Zeroual, I., Goldhahn, D., Eckart, T. and Lakhouaja 2019). containing 3.5 million news articles from 24 Arab countries, and a 1.5 billion words corpus from El-Khair (2016), derived from 5 million articles originating from 10 news sources.

### **2.3.2. T5 Model**

Raffel et al. (2020) introduce the Text-to-Text Transfer Transformer model, abbreviated as T5. T5's fundamental concept revolves around treating every language-related task as a "text-to-text" challenge. In essence, T5 takes textual data as input and generates new textual data as its output. T5 employs a unified approach to multi-task learning, utilizing the

same model, objective, training procedure, and decoding process across various English natural language processing tasks. These tasks encompass question answering, document summarization, machine translation, and sentiment classification. T5 adopts an architecture akin to an encoder-decoder Transformer with certain structural adjustments. Notably, it incorporates a normalization layer before a sub-block and introduces a pre-norm mechanism, which involves the initial input being part of the sub-block output. Both the encoder and decoder within T5 exhibit similar configurations and sizes as those found in BERT Base (Devlin et al. 2019). During training, T5 is exposed to the vast "Colossal Clean Crawled Corpus" (C4 for short), a dataset comprising approximately 750 GB of clean and naturally occurring English text extracted from the CommonCrawl web data. T5 demonstrates state-of-the-art performance across numerous benchmark assessments (Raffel et al. 2020).

### **2.3.3. XLM-RoBERTa**

RoBERTa shares similarities with BERT but boasts a more finely tuned set of hyperparameters. Expanding upon this, Conneau et al. (2020) extended RoBERTa's capabilities into the multilingual domain, resulting in XLM-RoBERTa (XLM-R). XLM-R was meticulously trained on an extensive dataset of over 2TB of filtered CommonCrawl data, spanning 100 different languages. This model is available in two variations, each with distinct specifications; XLM-RoBERTa Base: Comprising 12 layers, a hidden size of 768, 12 self-attention heads, a vocabulary size of 250K, and a total of 270 million parameters. XLM-RoBERTa Large: Featuring 24 layers, a hidden size of 1,024, 16 self-attention heads, a vocabulary size of 250K, and a total parameter count of 550 million. (Conneau et al. 2020)

It's noteworthy that both XLM-R models employ a similar masking objective as BERT but do not incorporate the next sentence prediction objective utilized in BERT. Conneau et al. (2020) have reported state-of-the-art performance across various tasks, including sequence labeling, cross-lingual classification, and question answering.

ALBERT In the work of (Li, Wang & Ding 2020), ALBERT, standing for "A Lite BERT for self-supervised learning of language representations," is introduced. ALBERT shares its foundation with other Transformer-based neural network architectures like BERT, RoBERTa, and XLM-R. However, it introduces two parameter reduction techniques to enhance training efficiency and reduce memory consumption compared to BERT (Devlin et al. 2019).

The first technique involves factorizing the embedding parameterization by decomposing the vocabulary embedding matrix into two smaller matrices. ALBERT employs input-level embeddings with a relatively lower dimensionality (e.g., 128) and hidden-layer embeddings with higher dimensions (e.g., 768, like BERT). The second technique is cross-layer parameter sharing, which applies the same layer repeatedly, leading to a significant reduction in the number of parameters while preserving performance.(Li, Wang & Ding 2020)

For instance, the ALBERT Base model comprises only 12 million parameters, representing an 89% reduction compared to the ALBERT Large model, which is similar in size to BERT Base but still performs impressively across three natural language understanding benchmarks: GLUE, SQuAD, and RACE. Moreover, it demonstrated that ALBERT Large can be trained approximately 1.7 times faster while utilizing roughly 18 times fewer parameters. (Li, Wang & Ding 2020).

#### 2.3.4. AraGPT2

GPT (Generative Pre-Trained Transformer), developed by Open AI, is a series of neural language models. Unlike BERT, GPT adopts an autoregressive approach, predicting the next word in a sequence based on context. GPT employs transformer architecture with decoder blocks, focusing on text generation rather than predicting subsequent words. The original GPT-1 model, trained on the Books Corpus dataset, utilized a 12-layer decoder with masked self-attention to prevent access to subsequent words during training. GPT-3 excelled in NLP tasks but exhibited limitations in natural language inference, coherence in long sentences, token weighting, and interpretability. Addressing these issues may involve enhancing training objectives, employing reinforcement learning, and considering additional modalities. Despite its capabilities, GPT models, including GPT-3, face challenges like generating biased content.

For the Arabic language, Antoun, Baly & Hajj (2023) introduced the first refined model, AraGPT2, exclusively developed on a large corpus of Arabic Internet texts and news articles. Their larger model, ARAGPT2-MEGA, with 1.46 billion parameters, has exhibited success in tasks such as synthetic news generation and zero-shot question-answering. The best AraGPT2-MEGA model achieved a perplexity of 29.8 on held-out Wikipedia articles, and human evaluation confirmed its proficiency in creating news articles indistinguishable from human-written ones. However, in a study Farha & Magdy (2021) evaluating 24 models for sentiment and sarcasm detection in Arabic texts, AraGPT2 variants showed comparatively lower performance, especially when compared to BERT-based models, indicating potential limitations in GPT-2 architecture for Arabic text-classification tasks.

## 2.4. Performance Evaluation

The evaluation metrics used are Accuracy, Precision, Recall, and F1-macro. A comprehensive assessment of a system's performance is often represented by the precision-recall curve, which is typically condensed into a single metric through the calculation of average precision across various standard recall levels or document counts. Other performance metrics, such as the break-even point or scaled utility as used in TREC, can also be defined to capture system effectiveness. However, one-dimensional synthetic performance measures fail to consider the inherent variability in scores, particularly when computed on limited data. It's important to note that this issue isn't exclusive to evaluations conducted on large collections. For instance, during the 2002 TREC filtering track, query 151 had just 22 relevant documents out of a total of 723,141 test documents. This means that a change in the classification of just one of those 22 relevant documents results in a substantial 5% variation in recall. (Gaussier & Goutte 2005).

For the sake of clarification, let's consider a straightforward scenario: each item is linked to a binary label, denoted as 'L' which signifies whether the item is correct or relevant to the given task. Furthermore, the system generates an assignment 'z' that indicates whether it deems the item as correct (relevant) or not. The experimental results can be conveniently summarized in a confusion table, see Table 3, where '+' and '-' represent relevant and non-relevant, 'TP' (true positive) and 'TN' (true negative) signify correct assignments, and 'FP' (false positive) and 'FN' (false negative) indicate incorrect assignments. With these counts, it becomes possible to calculate precision ( $p$ ) and recall ( $r$ ). (Gaussier & Goutte 2005)

**Table 3: Confusion Matrix**

|           |   | Assignment 'z' |    |
|-----------|---|----------------|----|
|           |   | +              | -  |
| Label 'L' | + | TP             | FN |
|           | - | FP             | TN |

Abstractly, accuracy refers to the proportion of correct predictions made by a model, expressed as a decimal value ranging from 0 (indicating no correct predictions) to 1 (indicating all predictions are correct) which is formulated in equation 1. In terms of probability, there is a natural explanation for both precision and recall. In fact, recall is the likelihood that a relevant object will be returned which is formulated in equation 2, whereas precision may be defined as the probability that an object is relevant given that it is returned by the system which is formulated in equation 3. (Gaussier & Goutte 2005)

The F1 score is a metric that combines both recall and precision, using their harmonic average. Its value falls between 0 and 1 which is formulated in equation 4. (M & M.N 2015) F1 macro is a performance metric used in multi-class classification tasks. It is the average of F1 scores for each class in the dataset. The F1 score for each class is calculated using precision and recall values (Harbecke et al. 2022). F1 macro takes values between 0 and 1, where a higher value indicates better performance which is formulated in equation 5.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$F1 - score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (4)$$

$$F1 - macro = (F1\ Score(1) + F1\ Score(2) + \dots + F1\ Score(n)) / n \quad (5)$$

Where:  $n$  is the total number of classes. True Positive (TP) refers to a group of tweets that have been correctly categorized as offensive. False Positive (FP) refers to a group of tweets that have been incorrectly categorized as offensive. True Negative (TN) refers to a group of tweets that have not been categorized as offensive and are correctly labeled as non-offensive. False Negative (FN) refers to a group of tweets that have not been categorized correctly and have been labeled as non-offensive or, but they belong to the offensive category.

## 2.5. Conclusion

The Technical Background chapter provides a comprehensive overview of classification techniques in text analysis. We examined the manual and statistical techniques, which involve the use of predefined queries and expert systems to categorize texts. We also explored machine learning techniques, which have become increasingly important for classifying large volumes of unstructured texts. Within the realm of machine learning, we discussed different types of neural networks, including FNN, CNN, RNN, and LSTM, and highlighted their applications in text classification tasks. Furthermore, we introduced the concept of transformers, which have gained popularity in natural language processing and

deep learning. Transformers offer efficient and effective ways to process and analyze textual data, enabling better classification accuracy and performance.

Overall, this chapter provides the necessary technical background and understanding of classification techniques in text analysis, setting the stage for the subsequent chapters where we apply these techniques to our specific problem domain. This chapter has equipped us with a solid understanding of the tools and methodologies available for text classification. As we move forward in this document, we will apply these techniques to address specific challenges and objectives in our research. Our exploration of these classification methods sets the stage for the practical implementations and analyses that follow, enabling us to harness the power of these tools to extract valuable insights from textual data. As we delve deeper into our research, we will build upon this foundation to achieve our goals and contribute to the broader field of text analysis.

## **Chapter 3: Literature Review**

The Literature Review chapter provides an in-depth analysis of relevant research and studies in the field of text classification (Wahdan, Al-Emran & Shaalan 2023). It begins by providing an overview of text classification, highlighting its significance in automating the categorization of documents based on their content. The chapter then focuses on Arabic text classification, discussing the unique challenges and complexities associated with the Arabic language, including its morphology and structure.

Furthermore, the chapter reviews existing studies on offensive language identification in Arabic, highlighting the various approaches and techniques employed in this area. It presents a summary of key findings and contributions from the literature, showcasing different models, preprocessing techniques, and ensemble methods used to improve the accuracy of offensive language detection.

The chapter also describes the systematic review methodology followed in the research, including the inclusion and exclusion criteria, search process, and quality assessment. It discusses the search period and databases used, as well as the number of publications initially retrieved, and the final number of studies included for analysis.

Additionally, the chapter presents the results of the systematic review, providing insights into the research domains and solution types in Arabic text classification. It explores trends and patterns in ATC research, emphasizing the areas of focus, such as healthcare, legal, security, social media, history, culture, and agriculture. The discussion highlights the

prevalence of social media content classification and the need for more research in the healthcare sector.

### **3.1. Overview**

The text classification process is the process of automating a set of documents into specific groups based on the content of the text itself through the use of certain technologies and algorithms (Elhassan & Ahmed 2015; Alabbas 2016; Salloum et al. 2017, 2018). Elhassan & Ahmed (2015) also defined the classification of texts as a method of searching for data and exploring them among large data and classifying them into groups for easy reference. According to Harrag (2009), the classification of texts was defined as the process of organizing documents according to a known and pre-existing structure of specific categories that suited this type of text. Others mentioned that there was a slight difference between text categorization and classification. Whereas the categorization of texts entailed sorting them according to their content, their classification placed them in certain groups that suited their content according to author, subject, title, language, and other classifications (Oo, Theeramunkong & Hung 2020). Specialized research on the classification of texts has increased significantly due to the enormous data available from many sources, including Internet pages, e-mail messages, news pages, texts circulated through social media, reports, and journal articles. Therefore, this research focused attention in order to make the best possible use of all these data and classify them (Li, Guo & Shang 2020).

### **3.2. Arabic Text Classification**

Arabic is one of the six primary languages worldwide. Its use is spread across many countries in the world, and it is the basis of the Arab world, but some Islamic countries also

use the Arabic language as their primary language because it is the language of the Qur'an. The Arabic language consists of 28 letters, including hamzas and elmodod. Each letter has certain structures that depend on the position of the letter in the whole, that is, whether it is at the beginning of the word, in the middle of the word, or at the end of the word (Abdeen 2019) . There are no upper and lower case letters in Arabic unlike English or other languages (Li, Guo & Shang 2020). The morphology of the Arabic language is not easy; it may be complicated as it has radical words, prefixes, and suffixes; moreover, unlike those in other languages, its words do not consist of sequential forms as its word structures are completely different and depend on position in the sentence and meaning (Li, Guo & Shang 2020).

As mentioned above, the classification of texts has become widespread, and most of the research includes the classification of other languages, such as English. Little of the research on the classification of texts has focused on Arabic. Many techniques and algorithms have been created to classify English texts, and they give excellent results and high accuracy due to the nature of the language, letters, and words. But when it comes to the Arabic language, the application of these algorithms does not give similar accuracy and validity in classification. Therefore, appropriate algorithms have been created to synthesize words in the Arabic language, but they have not yet demonstrated fully reliable merit for the classification process (Abdeen 2019).

### **3.3. Offensive Language Identification Methods**

The exponential growth of social media and online platforms has led to the proliferation of harmful and abusive content online, such as hate speech, cyberbullying, and

harassment. The detection and classification of offensive content have significant implications for social media platforms, content moderators, and society. By accurately identifying and removing offensive content, online platforms can create a safer and more inclusive environment for users. However, Arabic text classification presents unique challenges due to the richness of Arabic morphology, existence of various dialects in online contexts, and the lack of large standardized Arabic benchmark labeled dataset(Wahdan, Al-Emran & Shaalan 2023). Despite these challenges, researchers have made significant progress in developing effective Arabic text classification systems (Mubarak, Hassan & Chowdhury 2022a). Pretrained contextualized text representation models have revolutionized Natural Language Understanding (NLU) tasks and achieved remarkable performance in various NLP tasks (Devlin et al. 2019). Recently, many techniques have been used to improve the application of NLP, such as word embedding, deep learning, statistical methods, and graph representation (Wahdan, Salloum & Shaalan 2022).

In addition, there has been a growing interest in transfer learning approach, specifically finetuning large pretrained language models for downstream NLP/NLU tasks with limited training examples. This approach leverages the benefits of unsupervised language models. However, it has drawbacks, such as the necessity for large corpora for pre-training and the high computational resources necessary for training (Wahdan et al. 2020b). Furthermore, there is limited research on the combination of pretrained models, such as BERT, with neural networks in ensemble models, despite their potential for improved outcomes (Wahdan, Al-Emran & Shaalan 2023). Multilingual models have been created to circumvent this limitation by learning representations for multiple languages at the same time. However, these models have been predominantly available for English and a few other languages.

Arabic hate speech detection and offensive language identification have been the subject of several research papers. Table 4 provides a summary of key findings and contributions from the literature, with a focus on the OSACT dataset (Mubarak, Hassan & Chowdhury 2022b).

Mubarak, Hassan & Chowdhury (2022a) presented a baseline approach using SVM for offensive detection. They achieved an accuracy of 65.1% and an F1 score of 39.4%. They also experimented with different learning rates, using  $2e-5$  instead of  $8e-5$ . AlKhamissi & Diab (2022) proposed AraHS, a model for Arabic Fine-Grained Hate Speech Detection. Their approach outperformed the baselines, achieving an accuracy of 86% and an F1 score of 84.5%. They focused on predicting offensive language and fine-grained hate speech labels. Mostafa, Mohamed & Ashraf (2022) discussed their approach for the Offensive Language and Hate-Speech Detection shared task. They addressed the challenge of class imbalance and experimented with alternative loss functions and deep learning models. They used the Captum library for model interpretability. Their system's F1-score outperformed other models used in the shared task. Nessir et al. (2022) presented a multi-task model for hate speech detection on Arabic Twitter. They utilized contextualized text representation models such as MARBERT, AraBERT, and ARBERT. Despite the class imbalance issue, de Paula et al. (2022) explored different transformer models and ensemble methods for offensive detection. Their approach uses models, such as AraBERT, AraGPT2, and XLM-Roberta, which outperformed the baseline in all subtasks.

Makram et al. (2022) proposed a hybrid approach for the OSACT2022 shared task, utilizing MARBERT for feature extraction and employing classical machine learning

classifiers. Their Logistic Regression model achieved an accuracy of 80% and an F1 score of 78% for offensive tweet detection. Shapiro, Khalafallah & Torki (2022) addressed normalization techniques for Arabic hate speech detection, including Arabic letter normalization, punctuation normalization, digit normalization, hashtag segmentation, and diacritic removal. Their MarBERTv2 model achieved an accuracy of 85.6% and an F1 score of 84.1%. Elkaref & Abu-Elkheir (2022) used MARBERT's pre-trained word representations and applied a cluster-based approach to enhance semantic quality. They achieved an accuracy of 76.5% and an F1 score of 74.2% using a biLSTM model with k-means clustering. Alzu'bi et al. (2022) employed an ensemble model combining different pre-trained embeddings and linguistic features. Their approach achieved an accuracy of 86.4% and an F1 score of 84.9%. their system accuracy outperformed other models used in the shared task. They highlighted the challenge of normalizing dialects in offensive tweet detection.

In conclusion, the reviewed literature presents diverse approaches for offensive language identification in Arabic. These approaches employ different models, preprocessing techniques, feature extraction methods, and ensemble methods to enhance performance and address challenges like class imbalance and dialect normalization. It is evident that all the research papers have utilized pretrained models either as feature extractors or classifiers due to their effectiveness in capturing contextual information. Moreover, multiple pretrained models have been commonly utilized in the studies.

**Table 4: Performance comparison of offensive language identification methods**

| Paper                                           | Method used                                                                                       | Accuracy      | F1-score |
|-------------------------------------------------|---------------------------------------------------------------------------------------------------|---------------|----------|
| Baseline<br>(Mubarak, Hassan & Chowdhury 2022b) | Baseline SVM                                                                                      | 0.651         | 0.394    |
|                                                 | AraBERT                                                                                           | 0.820         | 0.800    |
|                                                 | QARiB                                                                                             | 0.840         | 0.823    |
| (AlKhamissi & Diab 2022)                        | AraHS                                                                                             | 0.860         | 0.845    |
| (Mostafa, Mohamed & Ashraf 2022)                | Ensemble (MARBERTV2 + MARBERT+ QARiB)                                                             | Not mentioned | 0.851    |
| (Nessir et al. 2022)                            | MARBERT fine-tuned with QRNN                                                                      | 0.854         | 0.839    |
| (de Paula et al. 2022)                          | AraBert                                                                                           | 0.841         | 0.827    |
| (Makram et al. 2022)                            | Marabert as feature extraction<br>Classification: logistic Regression                             | 0.800         | 0.780    |
| (Shapiro, Khalafallah & Torki 2022)             | MarBERTv2                                                                                         | 0.856         | 0.841    |
| (Elkaref & Abu-Elkheir 2022)                    | marabert as feature extraction<br>Classification: k-mean clustering then LSTM                     | 0.765         | 0.742    |
| (Alzu'bi et al. 2022)                           | Feature extraction: CBOW +Character level N-gram + word level N-gram TF-IDF embeddings<br>AraBERT | 0.864         | 0.849    |

### **3.4. Systematic Literature Review of ATC**

Several reviews were carried out on different ANLP applications, such as question answering (Alwaneen et al. 2022), sentiment analysis (Alhumoud & Al Wazrah 2022), text summarization (Al-Saleh & Menai 2016), and machine translation (Alqudsi, Omar & Shaker 2014). Concerning text classification, the existing literature witnessed several reviews that mainly focused on surveying methods and approaches. For instance, Alabbas et al. (2016) surveyed 48 studies published between 2006 and 2014 and categorized them by their main focus area, namely feature selection, term weighting, text classification algorithms, and stemming techniques. Farghaly and Shaalan (2009) covered the challenges that Arabic text classification (ATC) faced in a general ANLP sense and proposed some solutions that could guide the practitioners. Zahidi et al. (2019) focused on reviewing deep learning libraries used in ANLP. Wahdan et al. (2020) also conducted an intensive systematic review on ATC by reviewing deep learning approaches. The review focused on the type of neural network used to classify Arabic texts and discussed the evaluation criteria used in each study.

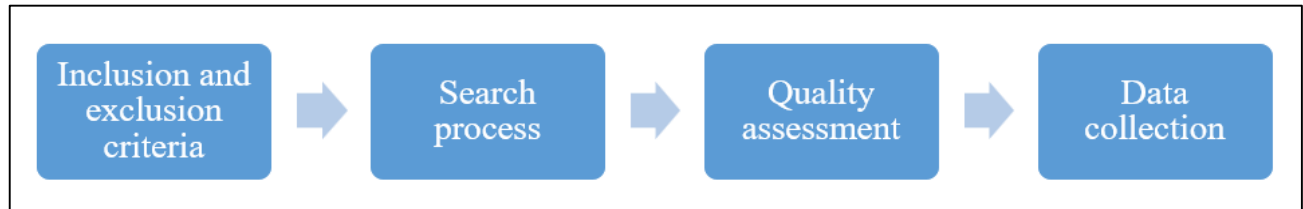
The preponderance of extant literature reviews predominantly concentrates on examining methodologies and approaches. Nonetheless, there has been no endeavor to review the trends and patterns in ATC research comprehensively. Understanding the current trends and patterns would enable the research community and practitioners to direct their investigations toward enhancing language processing and cultivating more dependable Arabic-based applications. The identified gaps motivated us to undertake an exhaustive systematic literature review to unveil contemporary ATC trends, patterns, and suggested solution modalities. This review further considers the research gaps encompassed by all trends and

patterns. Accordingly, this systematic review aims to answer the following research questions (Wahdan et al., 2023):

RQ1: What are the research domains of Arabic text classification?

RQ2: What are the proposed solution types in Arabic text classification?

This review follows the systematic review approach. This approach was introduced by Kitchenham et al. (2009). The technique comprises several stages, as demonstrated in Figure 8 and discussed in the following subsections.



**Figure 8: Systematic review approach**

#### **3.4.1. Inclusion and Exclusion Criteria**

Table 5 shows the criteria that we used to decide whether the study will be included or excluded in this review.

**Table 5: Inclusion and exclusion criteria**

| Inclusion criteria                                                                                                                                                                                                                                                                                                                                           | Exclusion criteria                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• It highlights ATC problems or weaknesses.</li><li>• It proposes a solution.</li><li>• Must report the model and the performance criteria used.</li><li>• It involves Arabic as a language for text classification.</li><li>• Must be written in English.</li><li>• It focuses on classifying digital text.</li></ul> | <ul style="list-style-type: none"><li>• Literature reviews and systematic reviews.</li><li>• It does not highlight any ATC problem or weakness.</li><li>• It does not propose any solution.</li><li>• It does not report the model and the performance criteria used.</li><li>• It involves languages other than Arabic for text classification.</li><li>• Not written in English.</li><li>• It does not focus on classifying digital text, such as focusing on photos, videos, or handwritten texts.</li><li>• Studies published by unknown publishers.</li></ul> |

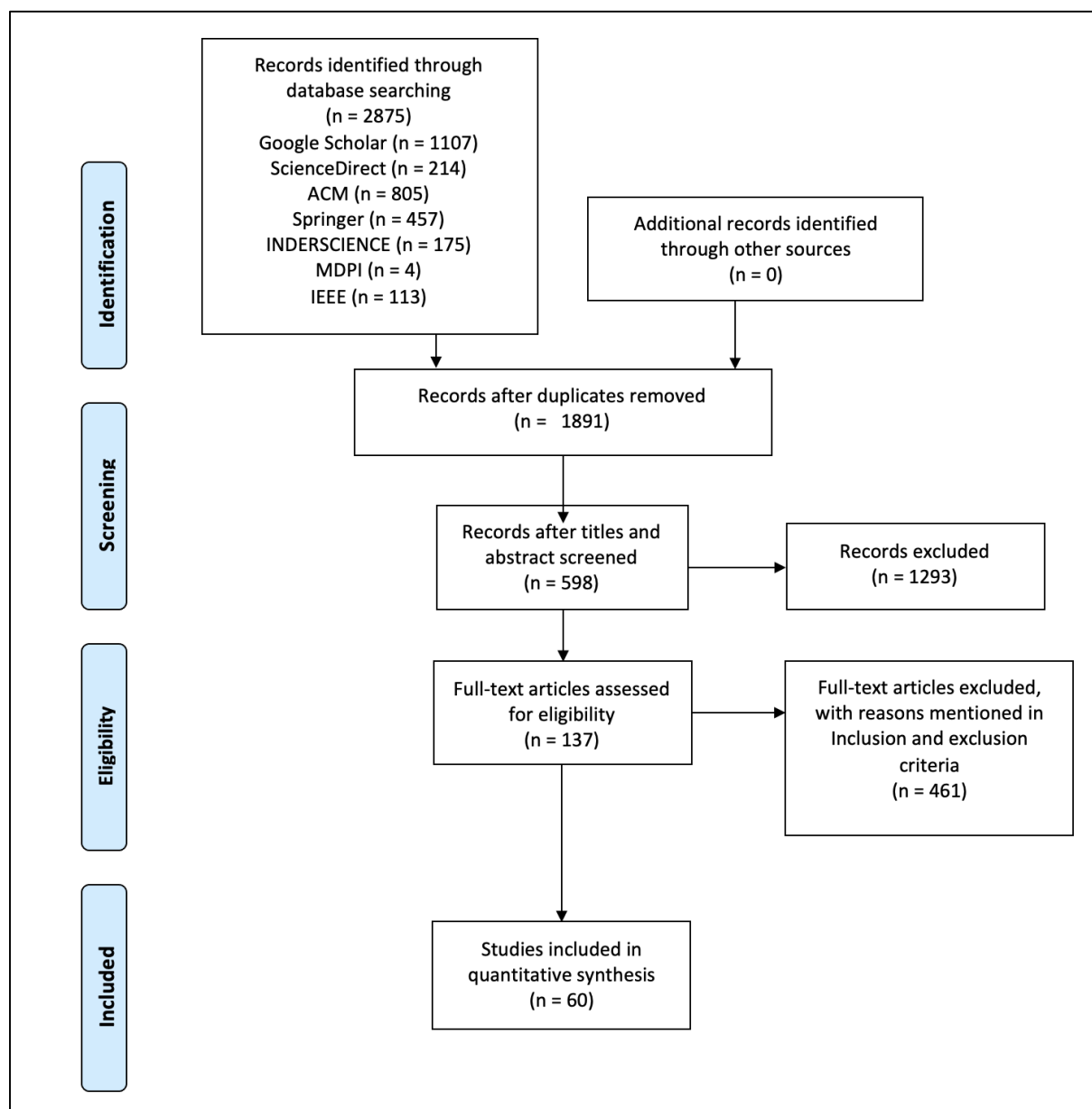
### **3.4.2. Search Process**

We performed our search through several online databases, including ScienceDirect, Springer, IEEE Explore, MDPI, ACM, and InderScience. We used several search strings in the search process, as detailed in Table 6. The search period is limited to be between 2017 to 2021. To cover all research published on Arabic text classification, we rephrased the term “text” with “document” and “classification” with “categorization”. The search in this systematic review is performed according to PRISMA (Preferred Reporting Item for Systematic Reviews and Meta-Analysis) (Moher et al. 2009). The initial number of publications retrieved is 2875. It is reduced to 1891 by removing duplicates. After screening titles and abstracts, 1293 publications were eliminated. We then applied the inclusion and exclusion criteria to each of the remaining studies. In this step, we excluded 461 publications

and ended up with 60 publications for further critical analysis. The PRISMA flowchart is shown in Figure 9.

**Table 6: Search criteria summary**

| No. | Category      | Details                                                                                                                 |
|-----|---------------|-------------------------------------------------------------------------------------------------------------------------|
| 1   | Keywords      | Arabic text classification; Arabic text categorization; Arabic document classification; Arabic document categorization. |
| 2   | Search string | ((Text OR Document) And (Classification OR Categorization) And Arabic)                                                  |
| 3   | Search period | 2017 – 2021                                                                                                             |



**Figure 9: PRISMA flowchart**

### 3.4.3. Quality Assessment

Each of the remaining studies ( $n = 60$ ) is evaluated according to the quality assessment (QA) criteria provided in Table 7. The questions were scored following a three-point scale. If the research answered the question thoroughly, it receives a 1; if it did not, it receives a 0, and

if it just partially responded to the question, it gets a 0.5. Table 8 exhibits the score of each study. It is apparent that all studies have passed the QA check.

**Table 7: Quality assessment questions**

| #   | Questions                                                          |
|-----|--------------------------------------------------------------------|
| QA1 | Is the text classification process described?                      |
| QA2 | Is the corpus identified well?                                     |
| QA3 | Is the study objective clearly identified?                         |
| QA4 | Did the author(s) make a comparison with other techniques?         |
| QA5 | Are the performance metrics clearly identified?                    |
| QA6 | Did the author(s) clearly highlight the contribution of the study? |

**Table 8: Quality assessment results**

| Study | QA1 | QA2 | QA3 | QA4 | QA5 | QA6 | Score (%) |
|-------|-----|-----|-----|-----|-----|-----|-----------|
| S1    | 1   | 1   | 1   | 1   | 1   | 1   | 100       |
| S2    | 1   | 1   | 1   | 1   | 1   | 1   | 100       |
| S3    | 1   | 1   | 1   | 1   | 1   | 1   | 100       |
| S4    | 1   | 1   | 1   | 1   | 0   | 1   | 83.4      |
| S5    | 1   | 1   | 1   | 1   | 1   | 1   | 100       |
| S6    | 1   | 1   | 1   | 0.5 | 1   | 1   | 91.7      |
| S7    | 1   | 1   | 1   | 1   | 1   | 1   | 100       |
| S8    | 1   | 1   | 1   | 1   | 0   | 1   | 83.4      |
| S9    | 1   | 1   | 1   | 1   | 1   | 1   | 100       |
| S10   | 1   | 0   | 1   | 1   | 1   | 1   | 83.4      |
| S11   | 1   | 1   | 1   | 1   | 1   | 1   | 100       |
| S12   | 1   | 1   | 1   | 1   | 1   | 1   | 100       |
| S13   | 1   | 1   | 1   | 1   | 1   | 1   | 100       |

|     |     |     |   |   |   |     |      |
|-----|-----|-----|---|---|---|-----|------|
| S14 | 1   | 0.5 | 1 | 1 | 1 | 1   | 91.7 |
| S15 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S16 | 0.5 | 1   | 1 | 1 | 0 | 0.5 | 66.7 |
| S17 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S18 | 1   | 1   | 1 | 1 | 0 | 1   | 83.4 |
| S19 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S20 | 1   | 1   | 1 | 1 | 0 | 1   | 83.4 |
| S21 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S22 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S23 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S24 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S25 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S26 | 1   | 0   | 1 | 1 | 1 | 1   | 83.4 |
| S27 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S28 | 0.5 | 0   | 1 | 1 | 1 | 1   | 75   |
| S29 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S30 | 1   | 1   | 1 | 1 | 0 | 1   | 83.4 |
| S31 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S32 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S33 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S34 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S35 | 1   | 1   | 1 | 1 | 0 | 1   | 83.4 |
| S36 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |
| S37 | 1   | 1   | 1 | 1 | 0 | 1   | 83.4 |
| S38 | 0.5 | 1   | 1 | 1 | 0 | 1   | 75   |
| S39 | 1   | 1   | 1 | 1 | 1 | 1   | 100  |

|     |     |     |     |   |   |     |      |
|-----|-----|-----|-----|---|---|-----|------|
| S40 | 0.5 | 1   | 1   | 1 | 0 | 0.5 | 66.7 |
| S41 | 1   | 1   | 0.5 | 1 | 1 | 1   | 91.7 |
| S42 | 1   | 1   | 1   | 1 | 1 | 1   | 100  |
| S43 | 1   | 1   | 0.5 | 1 | 1 | 1   | 91.7 |
| S44 | 1   | 0.5 | 0.5 | 1 | 1 | 1   | 83.4 |
| S45 | 1   | 1   | 1   | 1 | 0 | 1   | 83.4 |
| S46 | 1   | 1   | 1   | 1 | 1 | 1   | 100  |
| S47 | 1   | 1   | 1   | 1 | 1 | 1   | 100  |
| S48 | 0   | 0.5 | 1   | 0 | 1 | 1   | 58.4 |
| S49 | 0.5 | 1   | 1   | 1 | 1 | 1   | 91.7 |
| S50 | 0.5 | 0.5 | 1   | 0 | 1 | 1   | 66.7 |
| S51 | 1   | 0   | 1   | 1 | 1 | 1   | 83.4 |
| S52 | 0   | 1   | 1   | 0 | 1 | 0.5 | 58.4 |
| S53 | 1   | 1   | 1   | 0 | 1 | 1   | 83.4 |
| S54 | 0   | 0.5 | 1   | 1 | 0 | 1   | 58.4 |
| S55 | 1   | 1   | 1   | 1 | 0 | 1   | 83.4 |
| S56 | 1   | 1   | 1   | 1 | 1 | 1   | 100  |
| S57 | 0.5 | 0.5 | 1   | 1 | 1 | 1   | 83.4 |
| S58 | 1   | 1   | 1   | 1 | 1 | 0   | 83.4 |
| S59 | 1   | 1   | 1   | 1 | 1 | 1   | 100  |
| S60 | 1   | 1   | 1   | 1 | 1 | 1   | 100  |

#### 3.4.4. Data Collection

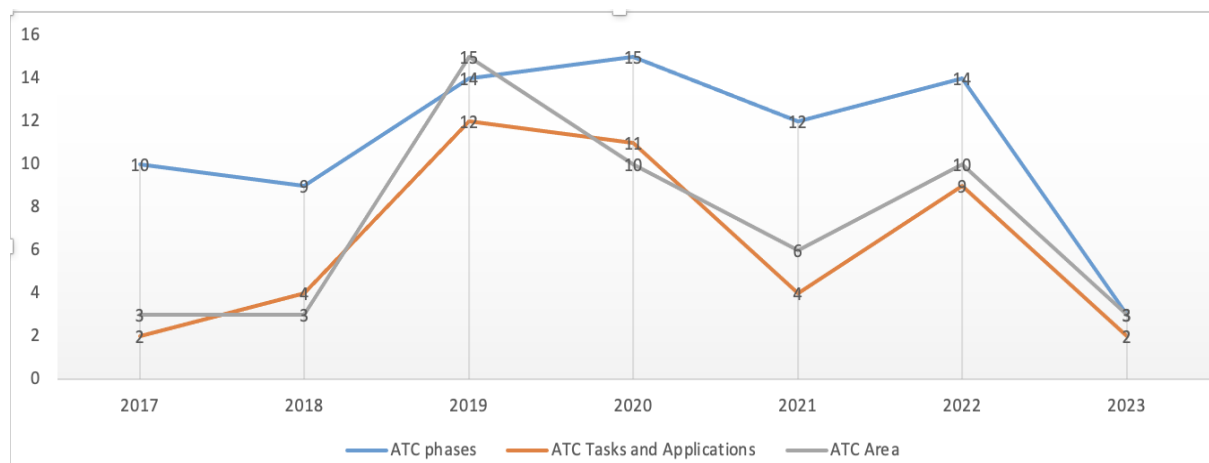
We have extracted the following characteristics from each study:

- The full reference of the papers and the source (i.e., journal or conference).
- The objective of the study.

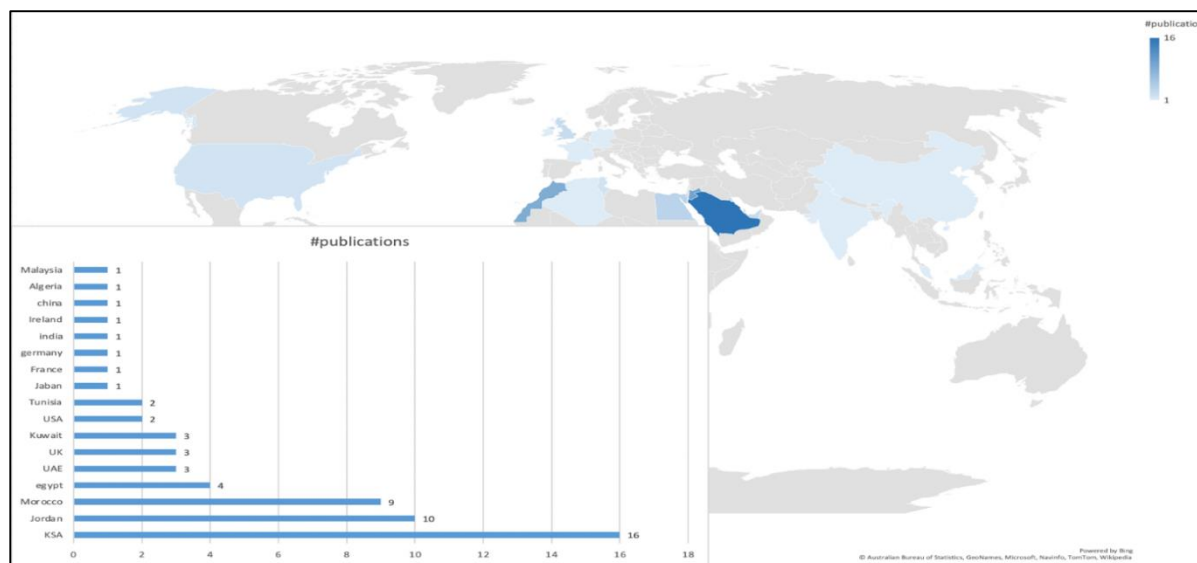
- The desired domain in ATC field of study.
- The application/ task that has been deployed using ATC (if any).
- The country where the authors are situated.
- Performance evaluation metrics.
- Text classification phases that have been improved (if any).
- Text classification technique used.

### **3.5. Systematic Review Results**

In 2022 and 2023, the data from Figure 9 underscores a notable trend in Automatic Text Classification (ATC) research. The emphasis remains on enhancing ATC phases, with 14 publications in 2022 and a subsequent decrease to 3 publications in 2023. While the number of studies focusing on ATC tasks and applications saw a slight decrease from 9 in 2022 to 2 in 2023, the attention to ATC areas remained consistent with 10 publications in both years. This suggests a continued interest in refining the classification process, with a potential shift towards exploring broader applications in the field of Automatic Text Classification. Figure 11 shows the research distribution per country. KSA is the prominent country in publishing studies on ATC, with 16 publications. This is followed by Jordan ( $n = 10$ ), Morocco ( $n = 9$ ), and Egypt ( $n = 4$ ). In the following subsections, we have discussed the answers to each of the research questions in this systematic review.



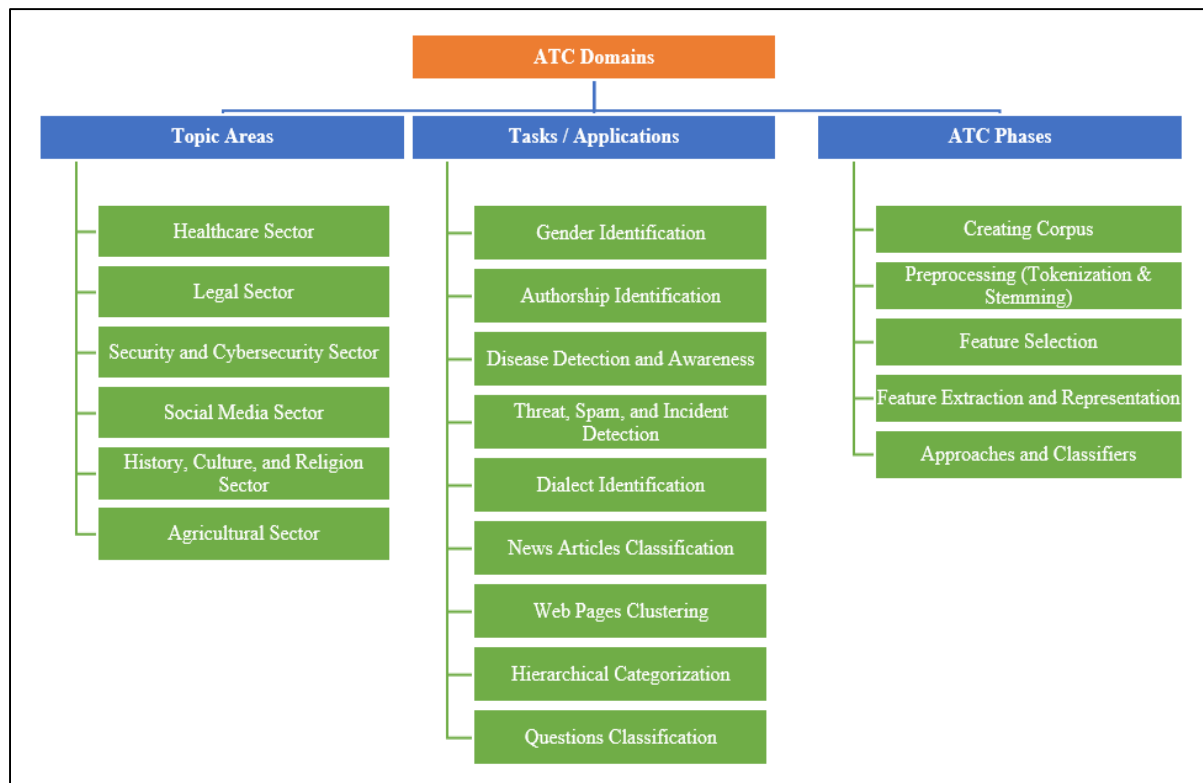
**Figure 10: Progress of publications in terms of domains and publication years.**



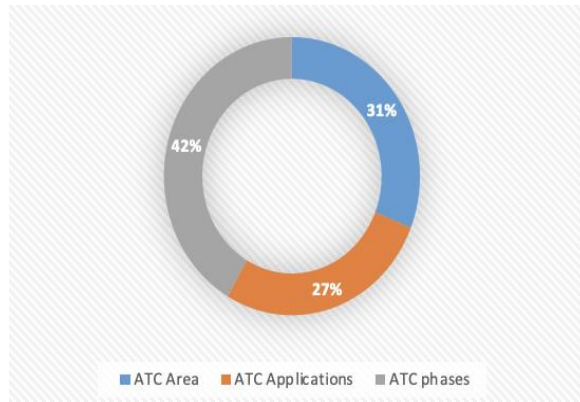
**Figure 11: Distribution of publications per country.**

### 3.5.1. Research Domains in Arabic Text Classification

Studies published on ATC have focused on different domains. We have classified them into three domains, including topic areas, tasks/applications, and ATC phases, as indicated in Figure 12. As it can be seen in Figure 13, 46% of publications focused on enhancing and improving ATC phases and processes, while 29% of them focused on ATC areas. 25% of the analyzed studies concentrated on ATC tasks and applications. The highest percentage of ATC phases indicates that improving the accuracy of ATC is still in its early stages and requires further attention from scholars.



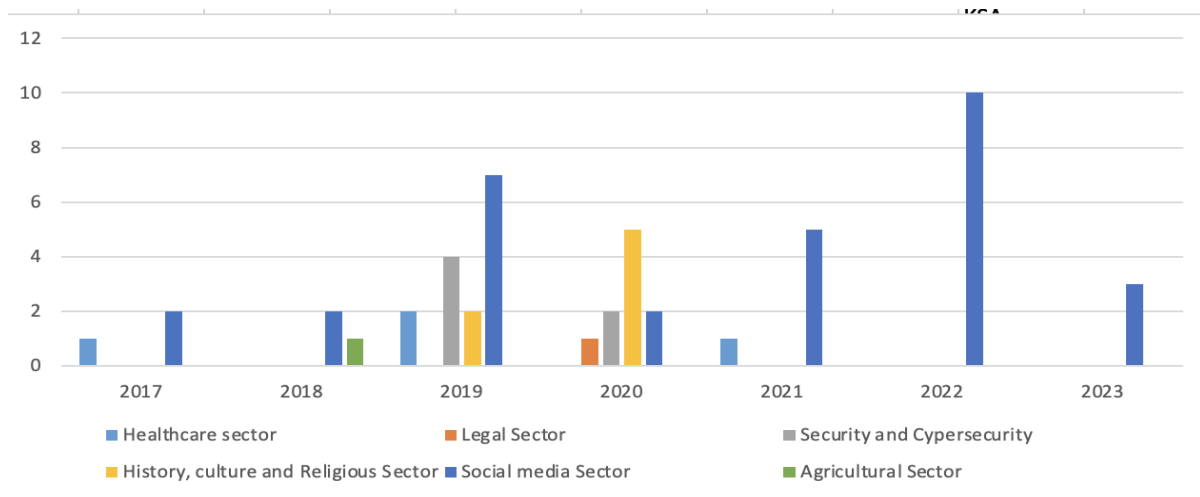
**Figure 12: ATC domains taxonomy**



**Figure 13: Distribution of ATC studies by domains**

### 3.5.1.1. ATC Topic Areas

Research publications on Automatic Text Classification (ATC) play a crucial role across various sectors, with studies categorized into six main sectors: healthcare, legal, security and cybersecurity, history, culture, and religion, social media, and agriculture. Figure 7 illustrates the distribution of publications across different years, while Figure 14 highlights trends in publications emphasizing topic areas. Notably, the total number of publications focusing on topic areas increased sharply in 2019, declined in 2020 and 2021, indicating decreased attention to ATC topic areas. Studies in 2019 predominantly focused on classifying social media content, shifting to Arabic texts related to history, culture, and religion in 2020. The healthcare sector receives relatively less attention, suggesting a potential area for further research. Social media content classification consistently ranks high, with particular emphasis on Twitter data in 2021, 2022, and 2023, underscoring Twitter's significant impact on shaping Arabic social media content.



**Figure 14: Distribution of ATC topic areas by publication year**

#### 3.5.1.1.1. Healthcare Sector

People have become more comfortable to publish their data on social media, with no exceptions for health-related issues. People discuss their diagnoses, symptoms, medications, and treatment results on social media. This encourages researchers to use these data to improve healthcare services (Badaro et al. 2019). The emergence of text classification in the healthcare sector has improved health quality from different angles, such as treatment, diagnosis, and management. Many government organizations have released useful health information to the public (Rayson & Alsudiasl 2019). For instance, Alghamdi (2019) has integrated text analytical tools in health applications to facilitate online diagnoses and treatments without visiting a psychologist. Rayson and Alsudiasl (2019) have supported online monitoring for infectious diseases by providing reliable information to avoid acting on false information, especially in emergency cases. Alnemer et al. (2015) aimed to evaluate the validity of health-related tweets and build awareness about the importance of spreading correct health-related information. Alnemer et al. (2015) found that more than 50% of the health-

related information spread on Twitter is false. Another study aimed to build a corpus on health services for opinion mining purposes (Alayba et al. 2017).

#### **3.5.1.1.2. Legal Sector**

Classifying Arabic legal documents is very important for law professionals because it assists them in retrieving legal texts. Also, it allows protecting sensitive data through automatic archiving judgments (Ikram & Chakir 2019). Despite its importance in this topic area, only one study has been found on ATC in the legal sector. Ikram and Chakir (2019) conducted an ATC study and focused on classifying court rulings automatically.

#### **3.5.1.1.3. Security and Cybersecurity Sector**

ATC has been used as a tool in the security and safety domain. ATC has been used to enhance monitoring the incidents and emergency cases and send suitable equipment to the exact location by analyzing data collected from Twitter (Alkhatib, El & Shaalan 2019). ATC has also been used in cybersecurity to minimize cyber-threats by detecting them in Instagram and Twitter posts (Alanazi 2019). IC3 (FBI Internet Crime Compliant) admits that the losses in 2018 were around \$2.7 billion because of online crimes and data breaches. This may cause an issue in national security (Alanazi 2019). Therefore, previous research has executed author profiling to identify authors, gender, and demographics (Alanazi 2019).

Abdulaziz et al. (2020) applied ATC to classify Instagram contents. Some researchers focused on detecting online threats, such as spam detection (Alorini & Rawat 2019; Ghourabi, Mahmood & Alzubi 2020). Alanazi (2019) conducted a study on automatic gender identification by finding discriminators between males and females from different perspectives, such as lexical, psychological, and sociological. Hussein et al. (2019) proposed

an ATC solution to detect gender from tweets, aiming to identify fake identities to face cybercrimes. Abooraig et al. (2018) classified political comments and classified them under the security sector.

#### **3.5.1.1.4. Social Media Sector**

In urban cities, social media is considered a robust platform with a vast amount of data. This is due to the increasing number of smartphones and the enormous amount of content created by users (Alkhatib, El & Shaalan 2019). Social media content is vital for many people and applications. The content can be classified using Arabic text categorization methods (Alhawarat & Aseeri 2020). Social media content can be used for different purposes. For instance, it can enhance the monitoring of incidents and emergency cases and send suitable equipment to the exact location by analyzing data from Twitter (Alkhatib, El & Shaalan 2019). Data collected via social media may aid in the online surveillance of infectious diseases (Rayson & Alsudiasl 2019). However, social media can provide misleading information. This will have a negative effect if the data are related to healthcare issues. Researchers proved that more than 50% of the data are incorrect using text classification methods. Alnemer et al. (2015) used this result to spread awareness to the community to not rely on social media content. Several studies have used social media content to approach their objectives in ATC (Alayba et al. 2017; Bekkali & Lachkar 2017; Abdelaal et al. 2018; Alhozaimi & Almishari 2018; Alanazi 2019; Hussein, Farouk & Hemayed 2019; Lippincott, Mcnamee & Duh 2019; Alorini & Rawat 2019; Altamimi & Teahan 2019; Hegazi et al. 2021; Omar et al. 2021; Alharthi, Alhothali & Moria 2021a; El-Alami, Ouatik El Alaoui & En Nahnahi 2021a; Hadj Ameer & Aliane 2021; Alruily et al. 2023). Additionally, all studies mentioned in Table 4

exclusively focus on social Twitter data, emphasizing the significance of Twitter in shaping Arabic social media content. Mubarak, Hassan & Chowdhury (2022a) presented a baseline approach using SVM for offensive detection. They achieved an accuracy of 65.1% and an F1 score of 39.4%. They also experimented with different learning rates, using  $2e-5$  instead of  $8e-5$ . AlKhamissi & Diab (2022) proposed AraHS, a model for Arabic Fine-Grained Hate Speech Detection. Their approach outperformed the baselines, achieving an accuracy of 86% and an F1 score of 84.5%. They focused on predicting offensive language and fine-grained hate speech labels. Mostafa, Mohamed & Ashraf (2022) discussed their approach for the Offensive Language and Hate-Speech Detection shared task. They addressed the challenge of class imbalance and experimented with alternative loss functions and deep learning models. They used the Captum library for model interpretability. Their system's F1-score outperformed other models used in the shared task. Nessir et al. (2022) presented a multi-task model for hate speech detection on Arabic Twitter. They utilized contextualized text representation models such as MARBERT, AraBERT, and ARBERT. Despite the class imbalance issue, de Paula et al. (2022) explored different transformer models and ensemble methods for offensive detection. Their approach uses models, such as AraBERT, AraGPT2, and XLM-Roberta, which outperformed the baseline in all subtasks.

Makram et al. (2022) proposed a hybrid approach for the OSACT2022 shared task, utilizing MARBERT for feature extraction and employing classical machine learning classifiers. Their Logistic Regression model achieved an accuracy of 80% and an F1 score of 78% for offensive tweet detection. Shapiro, Khalafallah & Torki (2022) addressed normalization techniques for Arabic hate speech detection, including Arabic letter normalization, punctuation normalization, digit normalization, hashtag segmentation, and

diacritic removal. Their MarBERTv2 model achieved an accuracy of 85.6% and an F1 score of 84.1%. Elkaref & Abu-Elkheir (2022) used MARBERT's pre-trained word representations and applied a cluster-based approach to enhance semantic quality. They achieved an accuracy of 76.5% and an F1 score of 74.2% using a biLSTM model with k-means clustering. Alzu'bi et al. (2022) employed an ensemble model combining different pre-trained embeddings and linguistic features. Their approach achieved an accuracy of 86.4% and an F1 score of 84.9%. their system accuracy outperformed other models used in the shared task. They highlighted the challenge of normalizing dialects in offensive tweet detection.

#### **3.5.1.1.5. History, Culture, and Religion Sector**

ATC has tackled the history, culture, and religion domain. Poems, for instance, are one of the tools used to document both history and culture. Poems are different than the regular text due to their rhythmic structure. Several studies were carried out to deal with poems from the ATC perspective. For example, Al-shaibani et al. (2020) used ATC algorithms to classify poems. Daif et al. (2020) used “image-based character encoder” followed by a deep learning classifier to classify poems. Khedher et al. (2020) categorized historical Arabic texts available in the national libraries using word recognition and text alignment. For religious topics, AL-Sarem and Emara (2019) focused on classifying short texts (e.g., fatwa) of religious documents extracted from Dar Al-ifta ALMisriyyah website using “Writing Style Features” for classification. In an additional study, Al-sarem et al. (2020) continued their work by enhancing their previous model using the attribution classifier. Aljedani et al. (2020) have also classified religious data (fatwa) based on its category. This is achieved by assigning multi-label text classification for each fatwa, so each one will have multi-hierarchical labels until the

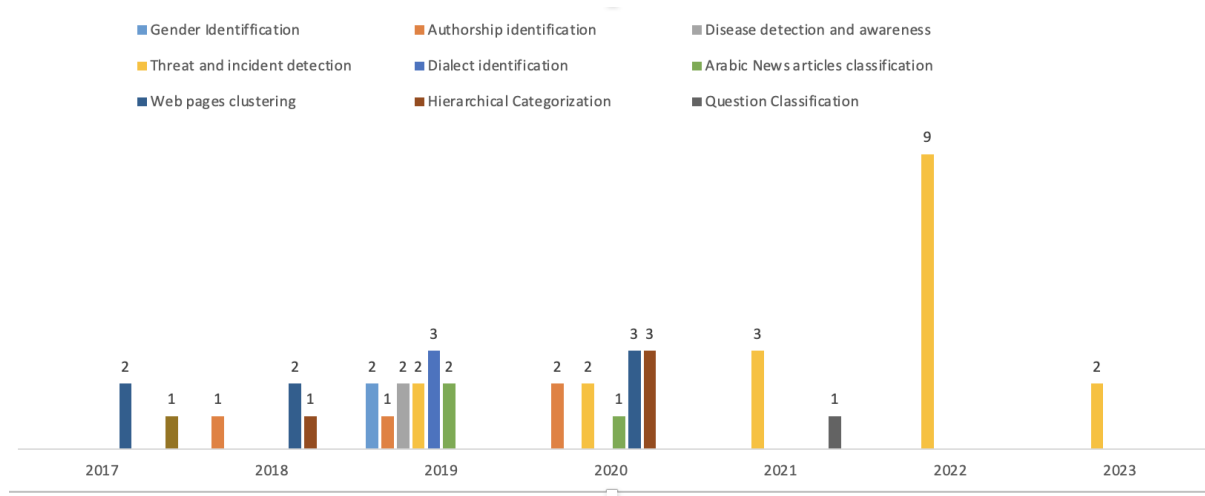
category reaches the general category. Hriez and Awajan (2020) have classified ancient books from a cultural perspective.

#### **3.5.1.1.6. Agricultural Sector**

The data analysis shows only one unique research under this category. In that, Guru et al. (2018) classified the textual complaints received from farmers.

#### **3.5.1.2. ATC Tasks and Applications**

At the core of numerous emerging applications is text categorization, a pivotal element in various industries(Elnagar, Al-Debsi & Einea 2020). As delineated in the preceding section, these applications have found utility across diverse sectors. Upon thorough literature review, our exploration identified nine primary task and application groups within current research, depicted in Figure 15. A total of 44 publications have concentrated on the utilization of Automatic Text Classification (ATC) for specific applications or tasks. Figure 14 provides an overview of the distribution of ATC applications and tasks across the years of publication, with detailed analyses of each category presented in the subsequent subsections.



**Figure 15: ATC tasks and applications across publication years.**

### 3.5.1.2.1. Gender Identification

Automatic gender identification of authors has attracted attention from practitioners due to its importance. This is attributable to the increasing number of fake identities and cybercrimes, such as child trafficking, sexual exploitation, terrorism, and other violence (Alanazi 2019). Alanazi (2019) aims to automate gender identification by finding discriminators between males and females by extracting features from different lexical, psychological, and sociological perspectives. The features were then fed into machine learning-based tasks. Hussein et al. (2019) have also proposed an ATC solution to detect gender from tweets. They employed feature selection methods such as n-gram and ensemble weighting for random forest and linear regression classifiers.

#### **3.5.1.2.2. Authorship Identification**

Authorship identification employs a predefined dataset for candidate authors to identify the author of the anonymous text (Hriez & Awajan 2020). This task attracts researchers due to the enormous volume of data available online. Several features can be found in the “Writing Style Features”, which can be categorized into different groups, including syntactic, semantic, lexical, structural, content and language-specific, and character (AL-Sarem & Emara2 2019). Hriez and Awajan (2020) identified the author by employing four features from the “Writing Style Features”, including lexical, semantic, syntactic, and structural features. They used the logistic model tree classifier to classify the texts. AL-Sarem and Emara (2019) classified short Arabic texts using three classifiers, including multilayer perceptron, Mahanobis distance, and linear regression using Stylometric features. They distinguished their research by studying the effect of increasing the training set size by mixing the “Writing Style Features”, a word-based lexical feature with n-gram and part-of-speech method. Al-sarem et al. (2020) employed the same approach of combining “Writing Style Features” with n-gram features. However, Al-sarem et al. (2020) aimed to classify short text documents and examine six attribution methods. Short text documents have been limited to 11 – 400 words, as determined by the authors. The used attribution methods involve decision tree C4.5, Markov model, K-NN, naive Bayes model, SMO, and Burrows Delta.

#### **3.5.1.2.3. Disease Detection and Awareness**

Alghamdi (2019) introduced a new approach by integrating a text analytical tool with a health application to detect depression and anxiety to support mental health. This approach relied on the technologies embedded in smartphones like the camera and self-test scales to

capture the health status. The application then diagnoses the patient using registered psychologists supported by text analytical tools to provide online services, such as advice, breathing exercises, and text-based therapy. Further, Alsudias and Rayson (2019) aimed to monitor false information and the spread of infectious diseases. It is believed that online health information must be reliable and high quality.

#### **3.5.1.2.4. Threat, Spam, and Incident Detection**

In emergency cases, one of the main challenges that face the rescued team is the lack of vital information about the incident, such as the location, number of injured citizens, number of vehicles, and the scale of the accident. A situational awareness about the incident lets rescued team decides the suitable equipment they need (Alkhatib, El & Shaalan 2019). Alkhatib et al. (2019) proposed a novel framework for monitoring incidents and events in smart cities. The framework aims to generate awareness and send warnings in emergency cases to the rescued team by collecting social media information and then generating an intelligent report. For threats, ATC has been used to detect online threats, such as spam detection. Ghourabi et al. (2020) used ATC to detect spam SMS messages, while Alharthi et al. (2021) detected low-quality content and accounts on Twitter. Further, Ameer and Aliane (2021) detected fake news and hate speech. This is similar to El-Alami et al. (2021), who detected offensive language. Alorini and Rawat (2019) also used ATC to detect spam and malicious news. Abdulaziz et al. (2020) claimed that there is a lack of research focusing on detecting threats from Arabic Instagram posts, and hence, they concentrated on classifying Instagram posts and contents. Arabic hate speech detection and offensive language identification have been the subject of several research papers mentioned in Table 4.

The exponential growth of social media and online platforms has led to the proliferation of harmful and abusive content online, such as hate speech, cyberbullying, and harassment. The detection and classification of offensive content have significant implications for social media platforms, content moderators, and society. By accurately identifying and removing offensive content, online platforms can create a safer and more inclusive environment for users. However, Arabic text classification presents unique challenges due to the richness of Arabic morphology, existence of various dialects in online contexts, and the lack of large standardized Arabic benchmark labeled dataset(Wahdan, Al-Emran & Shaalan 2023). Despite these challenges, researchers have made significant progress in developing effective Arabic text classification systems (Mubarak, Hassan & Chowdhury 2022a). Pretrained contextualized text representation models have revolutionized Natural Language Understanding (NLU) tasks and achieved remarkable performance in various NLP tasks (Devlin et al. 2019). Recently, many techniques have been used to improve the application of NLP, such as word embedding, deep learning, statistical methods, and graph representation (Wahdan, Salloum & Shaalan 2022).

#### **3.5.1.2.5. Dialect Identification**

Arabic dialect is the spoken Arabic language that differs from region to region. It is the unofficial language used daily between people more than the standard one (Altamimi & Teahan 2019). Recently, people have become more comfortable communicating using dialect language in written conversations and not only in spoken conversations. One of the recent phenomena in text categorization is the dialects categorization of social media posts (Altamimi & Teahan 2019). Lippincott et al. (2019) have researched dialect categorization using the

prediction by partial matching (PPM) model. They have tested improving the performance of dialect categorization using PPM and ensemble model, which consists of CNN, RNN, and MLP over character and word sequences. It has been observed that ensemble model could not beat the other model. Altamimi and Teahan (2019) also used PPM to categorize the text embedded in the dialect of the tweets. They have employed character and word-based approach for feature selection and PPM for tweet categorization. Altamimi and Teahan (2019) proved that PPM with a character-based approach outperformed three machine learning techniques, including KNN, Naïve Bayes (MNB), and Support Vector Machine (LIBSVM). Altamimi and Teahan (2019) differentiated their research by collecting data from six Arabic dialects, namely classic, modern, Levantine, Gulf, Maghrebi, and Egyptian. It has been noticed that modern and classic Arabic dialects have a close relationship. The Gulf and Iraqi dialects were also found to share the same characteristics. In contrast, Egyptian, Levantine, and Maghrebi dialects were easy to categorize and detect, as they have different features. The results of both studies (Altamimi & Teahan 2019; Lippincott, Mcnamee & Duh 2019) provided evidence that character-based feature extraction with PPM outperformed other techniques.

#### **3.5.1.2.6. News Articles Classification**

Arabic news articles are essential for many people and applications. To get the most benefit from them, they need to be organized and grouped. Arabic text categorization methods can thus be used for this task (Alhawarat & Aseeri 2020). Chantar et al. (2019) classified the news into 4 to 5 classes, but the main purpose of the research was to enhance text classification performance. Alhawarat and Aseeri (2020) have categorized news documents

using multi-kernel CNN. Madhfar and Al-Hagery (2019) used a corpus comprising Arabic documents that fall into five categories and classified these documents using classification models.

#### **3.5.1.2.7. Web Pages Clustering**

Web pages have high importance for many people and applications. Several Arabic text categorization studies were conducted to organize the content of web pages using different methods (Al-badarneh et al. 2017; Bahassine, Madani & Kissi 2017; Abooraig et al. 2018; Salama, Youssef & Fahmy 2018; Almuzaini & Azmi 2020; Elnagar, Al-Debsi & Einea 2020).

#### **3.5.1.2.8. Hierarchical Categorization**

Hierarchical categorization is the process of labeling the documents as a tree structure. The document, at a certain level, has a specific label and a parent label (Aljedani, Alotaibi & Taileb 2020; Almuzaini & Azmi 2020). Aljedani et al. (2020) have classified religious data to classify “fatwa” from the perspective of the category it belongs to. Aljedani et al. (2020) employed hierarchical text classification for each fatwa, so each one will have multi-hierarchical labels until the category reaches the general category. Almuzaini and Azmi (2020) proposed a model to classify Arabic texts with predefined multilevel labels – hierarchical categorization. Some other studies also used multi-labeling for texts (Al-anzi & Abuzeina 2018; Elnagar, Al-Debsi & Einea 2020).

#### **3.5.1.2.9. Questions Classification**

Questions classification aims to classify a given input question into predefined categories. While analyzing the collected studies in this review, this task appears only in one

study (Faris et al. 2021). It is an automated question classification for a question answering system. Faris et al. (2021) proposed a deep learning approach using LSTM and BiLSTM models to classify medical questions into 15 categories.

### **3.5.1.3. ATC Phases**

Several studies were carried out to enhance specific phases in the whole journey of ATC processing, as shown in Figure 5. However, each study has focused on specific phase/phases to enhance the performance. This section sheds light on the ATC phases from the perspective of the analyzed studies. Generally, most of the studies have focused on improving the approaches/ classifiers, with less attention paid to creating a new corpus and enhancing the preprocessing phase.

#### **3.5.1.3.1. Creating Corpus**

NADiA and SANAD are two large corpora that have been created by Elnagar et al. (2020). These two corpora were made open source to serve the research community. These two Arabic datasets were collected from the News portal. The Arabic poetry (AraP) and the Arabic Wikipedia title (AWT) are another two datasets that were created by Daif et al. (2020). The authors aimed to develop ATC benchmarks that do not need manual annotation. Chouigui et al. (2017) have developed an ANT corpus of news articles from RSS feeds.

A corpus is an extensive collection of text documents used to train and evaluate text classification models (Al\_Janabi, Salman & Mohammed 2020). Some studies have focused on studying the effect of the corpus size. For instance, AL-Sarem and Emara (2019) aimed to investigate the effect of changing the training dataset size in a specific field, while other studies have focused on examining the effect of increasing the size of the dataset itself

(Madhfar & Al-Hagery 2019; Omar et al. 2021). The latter two studies assured experimentally that increasing the dataset size has a high impact on the performance where it becomes more elevated and more stable. For gender identification purposes, Alanazi (2019) created the SDTwittC corpus from tweets, consisting of four million words, while Hussein et al. (2019) created the EDGA from 70,000 tweets. Table 9 shows a summary of the developed corpora in the existing literature.

Many studies have faced the unbalanced dataset issue. Hence, using machine learning for classification requires a balanced dataset to enable efficient learning and prediction (Al-Janabi et al. 2015; Mahdi & Al\_Janabi 2020). However, this issue has rarely been discussed in the ATC literature since the data being collected are not balanced, and the categories are not equal. This phenomenon is considered a major obstacle in text classification (Alayba et al. 2017; Almuzaini & Azmi 2020). While some recent solutions were proposed, more research is still required. For example, Daif et al. (2020) proposed a solution to deal with the class imbalance problem for text classification tasks using the CB SoftMax loss. Besides, Alsudias and Rayson (2019) suggested resampling the data using under-sampling using RUS and oversampling using ROS. In contrast, Madhfar and Al-Hagery (2019) claimed that there is no need for the sampling technique even if the dataset is unbalanced. One of the major issues that has been observed while analyzing the existing literature is the absence of an Arabic benchmark dataset to be used for comparing the models of ATC. This issue raises the need for developing an open-source benchmark dataset.

**Table 9: Summary of corpora in the existing literature**

| Source                                         | Corpus name                  | Data source                                                                          | Corpus size                                          | Type                                     |
|------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------|------------------------------------------------------|------------------------------------------|
| (Omar et al. 2021)                             | Not specified                | Facebook and Twitter                                                                 | 30,000 from Facebook and 14,000 from Twitter         | Modern standard Arabic                   |
| (Elnagar, Al-Debsi & Einea 2020)               | SANAD                        | News portal: Arabiya, Akhbarona Khaleej                                              | 200k articles                                        | Modern standard Arabic                   |
|                                                | NADiA                        | SkyNews Arabia and Masrawy                                                           | 485k articles                                        |                                          |
| (Daif, Kitada & Iyatomi 2020)                  | Arabic poetry (AraP)         | Adab Website and Arabic poetry from the 6th to 21st                                  | 444,911 different titles + 4,196,127 different words | Modern standard Arabic                   |
|                                                | Arabic Wikipedia title (AWT) | Arabic Wikipedia                                                                     | 41,264 poems                                         | Colloquial and old Arabic                |
| (Chouigui, Khiroun & Elayeb 2017)              | Arabic News Texts (ANT)      | News articles from Rtunisian radio website “Jawhara FM”                              | 6,005 articles                                       | Modern standard Arabic                   |
| (AL-Sarem & Emara2 2019)                       | Not specified                | Dar Al-ifta AL Misriyyah website                                                     | 4,631 text documents                                 | Not specified                            |
| (Madhfar & Al-Hagery 2019)                     | Not specified                | News sites: Hespress, Akhbarona, Assabah                                             | 111,728 Arabic documents                             | Modern standard Arabic                   |
| (Alayba et al. 2017; Madhfar & Al-Hagery 2019) | Not specified                | Twitter                                                                              | 2,026 tweets                                         | Not specified                            |
| (Alanazi 2019)                                 | SDTwittC                     | Twitter                                                                              | 4 million words                                      | Saudi dialect                            |
| (Hussein, Farouk & Hemayed 2019)               | EDGA                         | Twitter                                                                              | 70,000 tweets                                        | Egyptian dialect                         |
| (Jarrar, Khalilia & Ghanem 2022)               | Wojood                       | Online articles, websites, Awraq, the Birzeit University digital Palestinian archive | 550,000 tokens                                       | Modern Standard Arabic (MSA) and dialect |
| (Koshiry et al. 2023)                          | Not Specified                | Twitter                                                                              | 31,836 tweets                                        | Not Specified                            |

#### **3.5.1.3.2. Preprocessing**

Preprocessing is a crucial step in text classification as it improves the quality of the text data by cleaning and transforming it into a format that machine learning algorithms can easily understand (Al-Janabi, Salman & Mohammad 2019). In the preprocessing phase, the corpus is cleansed by removing stop words, punctuations, numbers, and non-Arabic characters, and then normalized (Bahassine, Madani & Kissi 2017; Almuzaini & Azmi 2020). The next step is tokenization. Each word is stemmed, as explained in the next subsection. Hegazi et al. (2021) proposed a cleaning algorithm in an opposite way. This approach only selects the required text but does not remove the noise. Al-ghamdi, Al-khalifa & Al-Salman (2023) addresses the gap in Arabic dependency parsing by comparing nine models, highlighting AraBERTv2's consistent superiority.

#### **3.5.1.3.3. Tokenization**

The process of breaking text into a list of tokens is known as tokenization. These tokens might be complete words or sub-words, which are parts of words. In addition to limiting the vocabulary size, the ability to create sub-words allows for universal tokenization across languages. A decent tokenization strategy achieves a fair granularity balance between characters and white space tokens (Alyafeai et al. 2021). Alyafeai et al. (2021) proposed three tokenization algorithms for classification and sentiment analysis.

#### **3.5.1.3.4. Stemming**

Stemming is considered one of the core steps in preprocessing and data cleansing. It is a computational procedure that diminishes the words into their core roots or stems (Al-badarneh et al. 2017; Almuzaini & Azmi 2020). There are two main stemmers in Arabic,

namely root-based stemmer and light stemmer or stem-based stemmer. Dlight stemmer is a light-based stemmer proposed by Alshalabi et al. (2021). Alshalabi et al. (2021) focused on finding and removing infix patterns from long and short Arabic words according to various rules, such as word length and precise order of the stemming phases to extract the double, triple, and quadruple roots.

Several studies examined the impact of stemming on ATC. Almuzaini and Azmi (2020) studied the impact of stemming on textual classification using deep learning models and argued that a stem-based algorithm is better. Other studies investigated the impact of stemming on ATC using the naïve Bayes model (Al-badarneh et al. 2017; Atwan et al. 2021). Al-badarneh et al. (2017) claimed that full word-stemming achieved the maximum accuracy and assured that using stemming in ATC has many advantages, such as reducing corpus size. This leads to more storage utilization and enhances processing time (Al-Janabi 2021). Atwan et al. (2021) assured that using light stemming achieves better accuracy. Alkhatib et al. (2019) studied the effect of stemming on the classification accuracy and concluded that the accuracy is increased by around 3%, but the penalty was increased in the response time.

#### **3.5.1.3.5. Feature Selection**

Feature selection refers to the process of excluding irrelevant features and selecting appropriate features that would be included in the text classification algorithm. This phase is important for enhancing accuracy (Abdeen et al. 2019; Larabi Marie-Sainte & Alalyani 2020). Some studies mainly aimed to identify the features of the text to reach out to the best feature selection method. For instance, Alanazi (2019) created an inventory of features to be used in automatic classification. The study focused on stylometric and psychometric features for

gender classification. Hussein et al. (2019) enhanced text classification by mixing several feature selection methods to create an ensemble weighted average, such as Mixed Feature Vector (MFV), N-Gram Feature Vector (NFV), and different stylometric and Egyptian Arabic Dialect (EAD). El-Alami et al. (2020) employed Chi-square for ranking the best features. However, Alshaer et al. (2021) compared normal Chi-square with improved Chi-square and proved that the latter achieves better performance. Hijazi et al. (2021) proposed a combination between Chi-square and Artificial Bee Colony (ABC). The results showed that fewer features could result in a higher weighted F1-measure score. Marie-Sainte and Alalyani (2020) proposed the Firefly algorithm-based feature selection for ATC and claimed that this method outperformed the state-of-the-art methods in improving ATC accuracy. Similarly, Chantar et al. (2019) proposed the Grey Wolf Optimizer to enhance the feature selection process. In contrast, Abooraig et al. (2018) classified a political dataset, compared the stylometric feature approach with the TC approach, and proved empirically that feature selection reduces the accuracy.

#### **3.5.1.3.6. Feature Extraction and Representation**

In NLP applications, word embedding models become a dominant building block due to their ability to encode syntactic regularities and linguistic similarities (Salama, Youssef & Fahmy 2018). Arabic language has a complex morphology, and ignoring this structure degrades the performance (Salama, Youssef & Fahmy 2018; Daif, Kitada & Iyatomi 2020). Almuzaini and Azmi (2020) revealed that the effect of classifying texts using word2vec combined with stemming strategies had increased the efficiency from the perspective of F-score. For word2vec, Almuzaini and Azmi (2020) used skip-gram and CBOW. For a large

dimension vector, Almuzaini and Azmi (2020) argued that it's better to combine CBOW with a stem-based algorithm. However, it's better to combine a stem-based algorithm with skip-grams for a small dimension vector.

Salama et al. (2018) have incorporated morphological annotations to the embedding model to enhance the Arabic word embedding. It is argued that the proposed enhanced model outperforms the state-of-the-art Arabic word embedding. Guru et al. (2018) used a novel term weighting, which is the TCW-ICF. El-Alami et al. (2020) claimed that the current methods in Arabic text representation ignore explicit knowledge. Hence, El-Alami et al. (2020) enriched Arabic text representation and captured the explicit semantics by mapping bag-of-word (BOW) to bag-of-concept (BOC). Besides, Alahmadi et al. (2017) combined BOW and BOC, proving that this combination enhances the performance of ATC. Contrary to BOW and BOC, a graph-based representation such as Babel Net has been used to encode relationships between named entities (Gouiouez & Hadni 2017; Hadni & Gouiouez 2017).

Term frequency-inverse document frequency (TF-IDF) is one of the most common feature extraction methods used in text classification (Larabi Marie-Sainte & Alalyani 2020). Alhaj et al. (2018) found that TF-IDF outperforms the BOW when SVM is used to classify the text. However, recent studies showed that this method did not work properly with ATC due to its complexity (Larabi Marie-Sainte & Alalyani 2020). Other feature extraction methods that appear during the literature synthesis include Linear Discriminant Analysis (LDA) (Abuzeina & Al-anzi 2017) and Latent Semantic Indexing (Al-anzi & Abuzeina 2017).

Some studies examined the impact of increasing the number of features representing the dataset. For example, Madhfar and Al-Hagery (2019) found that the effect is limited when

the number of features ranges between 1,000 and 3,000. However, if the features selected are greater than 3,000, it has a limited impact on performance. In such a scenario, Madhfar and Al-Hagery (2019) recommended removing the rarely occurring words to speed up the performance. Ayed et al. (2017) studied the effect of feature types on performance using n-grams and stem grams. El-Alami and El Alaoui (2018) proposed combining word sense disambiguation and document embedding, while Bekkali and Lachkar (2017) have used a web search engine to represent the text. El-Alami et al. (2021) claimed that the previous studies ignored the contextual information and meaning within the text in their word embedding models. Thus, El-Alami et al. (2021) proposed a pre-trained ARABERT for contextual text representation, and the results showed that this approach accomplished a state-of-the-art performance results with 99% of F1 score and accuracy. Table 10 summarizes the feature extraction studies in ATC.

**Table 10: Summary of feature extraction studies in the existing literature**

| Source                             | Feature extraction methods                                        | Corpus                                                      |
|------------------------------------|-------------------------------------------------------------------|-------------------------------------------------------------|
| (El-Alami et al., 2021)            | Contextual semantic embeddings based on fine-tuned AraBERT model. | SemE- val'2020                                              |
| (Almuzaini & Azmi 2020)            | Word2vec – word embedding technique.                              | ANT and SPA                                                 |
| (Salama, Youssef & Fahmy 2018)     | Sense2vec, word2vec, and CBOW.                                    | ASTD-ArTwitter-QRCI<br>MPQA-Arabic                          |
| (Guru, Ali & Suhil 2018)           | TCW-ICF (Term Class Weight-Inverse Class Frequency).              | 5000 complaints of the farmers collected by the author      |
| (El-Alami et al., 2020)            | BOC and Chi-square.                                               | OSAC                                                        |
| (Alahmadi, Joorabchi & Mahdi 2017) | BOC and BOW.                                                      | Saudi News Papers (SNP) dataset,<br>Al-Khaleej, Arabic 1445 |

|                                                                                       |                                                                               |            |
|---------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|------------|
|                                                                                       |                                                                               | dataset    |
| (Gouiouez & Hadni 2017; Hadni & Gouiouez 2017)                                        | Babel Net                                                                     | EASC       |
| (Alhaj et al. 2018; Al-Radaideh & Al-Abrat 2019; Larabi Marie-Sainte & Alalyani 2020) | Term frequency-inverse document frequency (TF-IDF).                           | CNN Arabic |
| (Elkaref & Abu-Elkheir 2022)                                                          | Marabert as feature extraction<br>Classification: k-mean clustering then LSTM | OSACT      |
| (Makram et al. 2022)                                                                  | Marabert as feature extraction<br>Classification: logistic Regression         | OSACT      |

#### 3.5.1.4. ATC Approaches and Classifiers

Table 11 summarizes the ATC approaches and their corresponding algorithms in the analyzed ATC literature. The details of these approaches are discussed in the following subsections.

##### 3.5.1.4.1. Machine Learning

Machine learning assists in classification by automatically identifying patterns and relationships within datasets, enabling a model to categorize and predict outcomes based on input data (Zaza & Al-Emran 2015). In Arabic dialect identification, Altamimi and Teahan (2019) and Lippincott et al. (2019) explored the effect of one of the compression schema, namely Prediction by Partial Matching (PPM). Al-Radaideh and Al-Abrat (2019) aimed to enhance the efficiency of ATC by utilizing Binary Grey Wolf Optimizer (BGWO) as a wrapper feature selection. Rough set theory (RST) is another machine learning technique that has been used in text classification. RST involves building the decision table, reducing

generation, which can be either single or multiple reductions, and generating classification rules. Al-Radaideh and Al-Abrat (2019) generated multiple reductions in RST to classify the texts and claimed that multiple reductions outperformed the single reduction in their experiment. It is also argued that RST performance is enhanced with multiple reduction extraction.

VSM is another method that has been used for textual representation. Bahassine et al. (2017) proposed using VSM to represent the document by the weight of the terms and find out the similarity between the documents. VSM has also been used by deploying a hierarchical classification using the Markov chain (Al-anzi & Abuzeina 2018). However, Guru et al. (2018) contradicted the efficiency of VSM and stated that KNN is better after performing a comparative analysis. Cosine similarity is one of the most popular machine learning methods in ATC, and it has been used by Al-anzi and Abuzeina (2017). Other approaches used for classification include the Homer algorithm (Aljedani, Alotaibi & Taileb 2020) and the logistic model tree (Hriez & Awajan 2020).

#### **3.5.1.4.2. Deep Learning**

Deep learning has improved text classification by providing powerful techniques for automatically learning complex patterns and relationships in text data (Al-Janabi, Alkaim & Adel 2020; Al-Janabi, Mohammad & Al-Sultan 2020). Several studies have used the deep learning approach as a classifier. For instance, Alhawarat and Aseeri (2020) have categorized news documents using multi-kernel CNN. They called their model as “Superior Arabic Text Categorization Deep Model (SATCDM)”. They used “superior” due to the super efficiency of the proposed model that outperforms similar models. It has been argued that the proposed

model can work efficiently regardless of the preprocessing methods. Sundus et al. (2019) have used feed-forward deep learning in their text classification model. It has been noticed that using deep learning models is very promising in ATC as these models improved the performance significantly from the perspective of time and accuracy.

Deep learning has also proven its efficiency in classifying Arabic poems. Al-shaibani et al. (2020) used deep bidirectional recurrent NN for meter classification of poems, while Daif et al. (2020) used image-based character embedding in CNN. Besides, Elnagar et al. (2020) employed different deep learning models to prove their efficiency in classification without the preprocessing stage. Ghourabi et al. (2020) proposed a hybrid model by combining CNN and LSTM to enhance SMS classification. El-Alami et al. (2020) proposed a deep autoencoder classifier constructed of multiple layers of feed-forward NN. Employing this model in classifying applications is efficient because of its ability to reconstruct inputs without labeling. Faris et al. (2021) proposed a question classification model using unidirectional LSTM and bidirectional LSTM (BiLSTM) and proved that BiLSTM achieves better accuracy. Convolutional neural network (CNN) has also been used in ATC (Abdulaziz, Abdul & Jilani 2020; AlSaleh & Larabi-Marie-Sainte 2021). AlSaleh and Larabi-Marie-Sainte (2021) proposed genetic algorithm-based CNN to optimize the CNN parameter and improve the accuracy by around 5%.

#### **3.5.1.4.3. Transformer**

BERT, a powerful approach in NLP, has garnered attention due to its ability to predict words based on both left and right contexts and to be pre-trained using plain text corpora accessible on the web. To accommodate various languages, including Arabic, multiple BERT

models have been developed. Nevertheless, the current understanding and implementation of BERT models for Arabic text classification remain limited.

MARBERT is explicitly designed for the Arabic language and was trained on an extensive Twitter dataset, encompassing both Modern Standard Arabic and diverse Arabic dialects. The pretraining corpus comprises one billion tweets and nearly 128 gigabytes of text. The corpus contains approximately 15.6 billion tokens, roughly double the size of AraBERT Version 2, making it the most extensive pretraining corpus among the nine models. While MARBERT shares the same architecture as Multilingual BERT, it omits the next-sentence prediction (NSP) feature. The developers opted to exclude NSP due to the typically short length of tweets. The model possesses around 160 million trainable parameters (Abdul-Mageed, Elmadany & Nagoudi 2021).

MARBERT's developer also developed ARBERT, although there are notable differences between the two models. ARBERT was pretrained exclusively on Modern Standard Arabic text, collected from sources such as Wikipedia, news articles, and books, totaling approximately 61 gigabytes and 6.2 billion tokens. Its architecture resembles Multilingual BERT, featuring 12 transformer blocks, 768 hidden units, and 12 self-attention heads, but it has around 163 million trainable parameters (Antoun, Baly & Hajj 2020).

#### **3.5.1.4.4. Ensemble Approach**

Other approaches have combined more than one classifier to deploy ensemble models (Abdelaal et al. 2018; Lippincott, Mcnamee & Duh 2019). In that, Abdelaal et al. (2018) combined decision tree, NB, and sequential minimal optimization (SMO). Besides, Al-Tamimi et al. (2021) proposed an ensemble model to achieve active learning for ATC. It has been

demonstrated that the ensemble model achieves a significant improvement in the efficiency of machine learning compared to traditional passive learning.

Alruily et al. (2023) proposed a method for classifying lengthy Arabic tweets by employing transfer learning in conjunction with BERT. This approach involves fine-tuning a pre-trained BERT model on an extensive Arabic text corpus and then fine-tuning the model on a smaller labeled dataset of long Arabic tweets for a specific classification task. The authors also employ data augmentation techniques to expand the training dataset and enhance the model's performance. The classification task entails identifying tweets as either related or unrelated to a particular topic of interest. The authors assess the performance of their method using various metrics, such as precision, recall, F1-score, and accuracy. The results indicated that the proposed method accurately classifies long Arabic tweets related to the topic of interest. The observations from this study suggest that both neural networks and BERT exhibit exceptional performance and have established new benchmarks in diverse NLP tasks. Consequently, it is hypothesized that integrating these two techniques could significantly improve the accuracy of NLP models.

Incorporating new layers atop a pre-trained Transformer encoder is an emerging method in both research and industry for adapting pre-trained models to specific downstream tasks. This approach constitutes a form of transfer learning whereby the pre-trained model is fine-tuned on a new task using a smaller dataset, hoping that the pre-trained model has acquired valuable features that can be adapted to the new task (Devlin et al. 2019).

**Table 11: Summary of ATC approaches and their corresponding algorithms in the existing literature.**

| ATC approaches   | Sources                                                              | Corpus                                                                                 | Algorithms                           |
|------------------|----------------------------------------------------------------------|----------------------------------------------------------------------------------------|--------------------------------------|
| Machine learning | (Altamimi & Teahan 2019; Lippincott, Mcnamee & Duh 2019)             | Bangor Twitter Arabic corpus (BTAC)                                                    | Prediction by Partial Matching (PPM) |
|                  | (Al-Radaideh & Al-Abrat 2019)                                        | Available online (Www.diab.edublogs.org)                                               | Rough Set Theory (RST)               |
|                  | (Bahassine, Madani & Kissi 2017; Al-anzi & Abuzeina 2018)            | Collected from the Arabic online newspapers “Hespress” <sup>1</sup> and “Hesport       | VSM                                  |
|                  | (Guru, Ali & Suhil 2018)                                             | 5000 complaints of the farmers collected by the author                                 | KNN                                  |
|                  | (Aljedani, Alotaibi & Taileb 2020)                                   | Raw dataset related to the Islamic field and written in Modern Standard Arabic (MSA).  | Homer algorithm for classification   |
|                  | (Al-anzi & Abuzeina 2017)                                            | Alqabas newspaper                                                                      | Cosine similarity                    |
|                  | (Hriez & Awajan 2020)                                                | AAAT                                                                                   | Logistic model tree                  |
| Deep learning    | (Abdulaziz, Abdul & Jilani 2020; AlSaleh & Larabi-Marie-Sainte 2021) | 2000 Arabic Instagram collected by the author                                          | CNN                                  |
|                  | (Alhawarat & Aseeri 2020)                                            | CNN, BBC, OSAC, ALwatan, Aljazeera, AL Khaleej, Abuaiadah (V1), Abuaiadah (V2-5), NADA | Multi kernel CNN                     |
|                  | (Sundus, Al-Haj & Hammo 2019)                                        | Khaleej-2004                                                                           | Feed-forward deep learning           |
|                  | (Al-shaibani, Alyafeai & Ahmad 2020)                                 | Metrec dataset (Poetry) from Al-Shaibani et al., 2020), AJGT(tweets)                   | Deep bidirectional recurrent NN      |

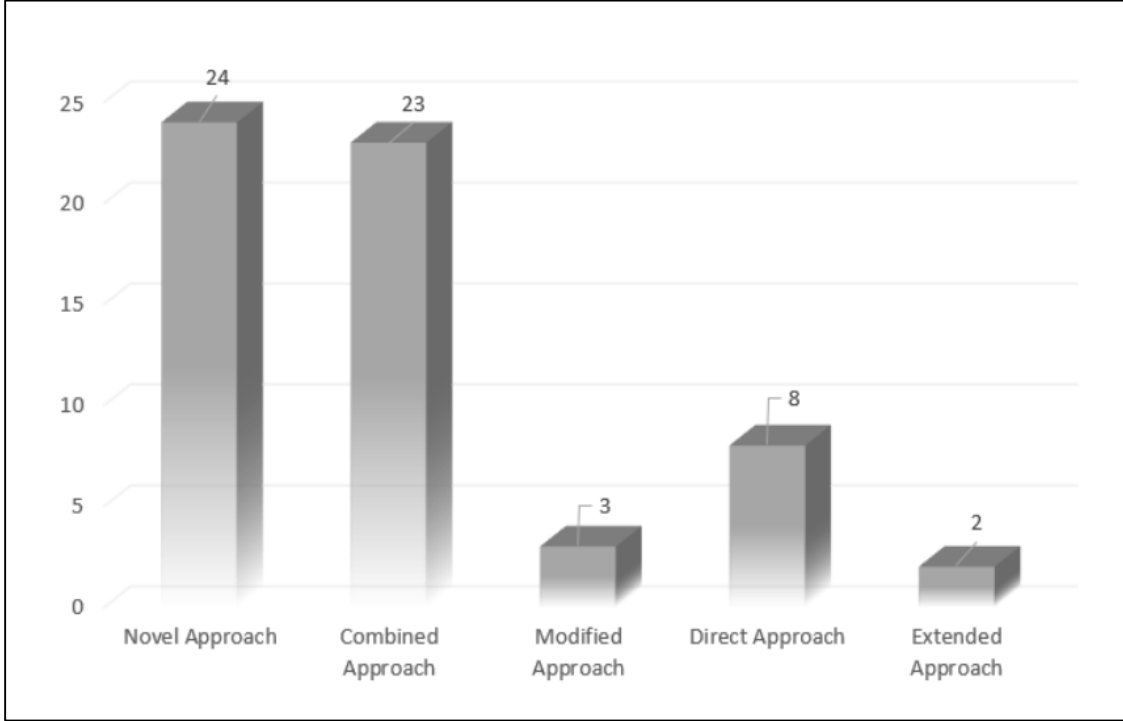
|                   |                                         |                                                                                                                       |                                                              |
|-------------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
|                   |                                         | 2017, LABR(Book), DSAC 2018                                                                                           |                                                              |
|                   | (Ghourabi, Mahmood & Alzubi 2020)       | SMS Spam dataset from the UCI Repository, and a set of Arabic messages collected from local smartphones by the author | CNN and LSTM                                                 |
|                   | (Faris et al. 2021)                     | Medical consultations from Altibbi company.                                                                           | BiLSTM                                                       |
|                   | (El-Alami et al., 2020)                 | OSAC                                                                                                                  | Multiple layers of feed forward NN                           |
| Transformers      | (Antoun, Baly & Hajj 2020)              | OSACT                                                                                                                 | Arabert                                                      |
|                   | (Abdul-Mageed, Elmadany & Nagoudi 2021) | ARCD, ASTD, ArSarcasm, GLUE, LABR, SuperGLUE, decaNLP                                                                 | Marabert                                                     |
|                   | (Almaliki et al. 2023)                  | The Arabic Hate-Speech (AHS)                                                                                          | Arabic BERT-Mini Model (ABMM)                                |
|                   | (Mubarak, Hassan & Chowdhury 2022b)     | OSACT                                                                                                                 | AraBERT, QARiB                                               |
|                   | (AlKhamissi & Diab 2022)                | OSACT                                                                                                                 | AraHS                                                        |
|                   | (Nessir et al. 2022)                    | OSACT                                                                                                                 | MARBERT fine-tuned with QRNN                                 |
|                   | (de Paula et al. 2022)                  | OSACT                                                                                                                 | AraBert                                                      |
|                   | (Shapiro, Khalafallah & Torki 2022)     | OSACT                                                                                                                 | MarBERTv2                                                    |
| Ensemble approach | (Abdelaal et al. 2018)                  | 500 tweets<br>Collected by author                                                                                     | Decision tree, NB, and Sequential Minimal Optimization (SMO) |
|                   | (Lippincott, Mcnamee & Duh 2019)        | ADC, LID                                                                                                              | CNN, RNN, and MLP-PPM                                        |
|                   | (Al-Tamimi, Bani-Isaa & Al-Alami 2021)  | NADA, OSAC                                                                                                            | SVM with stochastic gradient descent (SGD) learning method   |

|  |                                    |                                                                                                                      |                                                                                                 |
|--|------------------------------------|----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
|  |                                    |                                                                                                                      |                                                                                                 |
|  | (Alruily, Fazal & Mostafa 2023)    | OSIAN                                                                                                                | Transfer learning with BERT                                                                     |
|  | (Abdelgwad, Soliman & Taloba 2022) | Human Annotated Arabic Dataset of Book Reviews (HAAD), the Arabic news dataset, and the Arabic Hotel Reviews Dataset | AraBERT with sentence pair input                                                                |
|  | (Alzu'bi et al. 2022)              | OSACT                                                                                                                | Feature extraction: CBOW +Character level N-gram + word level N-gram TF- IDF embeddings AraBERT |
|  | (Mostafa, Mohamed & Ashraf 2022)   | OSACT                                                                                                                | Ensemble (MARBERTV2 + MARBERT+ QARiB)                                                           |
|  | Nessir et al. (2022)               | OSACT                                                                                                                | Combination of pretrained models with neural networks                                           |

### 3.5.2. Proposed Solutions in Arabic Text Classification

We have analyzed the collected ATC studies to identify the type of proposed solutions. The proposed solutions were categorized into five different categories. These categories include novel (original) approach, direct (existing) approach, combined approach (a combination of existing approaches), extended approach (enhancing an existing approach by adding a supplement solution), and modified approach (enhancing an existing approach with a minor modification). Figure 16 shows the number of studies under each category. It can be noticed that most of the analyzed studies have proposed a novel approach, with 24 publications. This is followed by those that proposed a combined approach (n = 23), direct approach (n = 8), modified approach (n = 3), and extended approach (n = 2). These results indicate that recent studies are focusing on proposing new solutions, specifically with the

advent of machine learning and deep learning techniques. At the same time, more research is required to solve the challenges in ATC, specifically those arising from the language complexity in comparison to other languages.



**Figure 16: Proposed solutions in ATC.**

### **3.6. A Comparative Analysis of this Systematic Reviews and Previous Review**

In Table 12, we compare our systematic review with previous reviews in terms of their advantages and disadvantages. Each previous review offers unique contributions and focuses on specific aspects of the Arabic language processing domain. We summarize the advantages and disadvantages of each review to provide a comprehensive understanding of the landscape and highlight the distinctiveness of our systematic review.

It is important to note that our systematic review stands out by offering a comprehensive analysis of 60 research articles, covering a wide range of topics, applications, methods, and techniques in Arabic text classification (ATC). We organize the gathered studies into three main categories: topic areas, tasks/applications, and ATC phases. This categorization enables a thorough examination of the current state-of-the-art in ATC and provides valuable insights for future research endeavors. Additionally, our review emphasizes the proposed solutions presented in each ATC study, contributing to a holistic understanding of ATC's full potential.

Furthermore, we acknowledge that each previous review has its own advantages and disadvantages. For example, Alwaneen et al. (2022) provide an in-depth examination of contemporary Arabic Question Answering Systems (QASs) and categorize recent studies based on the types of questions addressed. Alhumoud & Al Wazrah (2022) focus on sentiment analysis and employ keyword co-occurrence analysis to reveal prevailing hotspots and temporal trends. Al-Saleh & Menai (2016) offer a comprehensive review of Arabic text summarization techniques, while Alqudsi, Omar & Shaker (2014) concentrate on machine translation methodologies. Alabbas, Al-Khateeb & Mansour (2016) aim to identify and summarize various techniques employed in classifying Arabic text, and Farghaly & Shaalan (2009) discuss challenges in Arabic Natural Language Processing (ANLP) applications. Wahdan et al. (2020a) analyze recent studies on text classification in Arabic using neural network algorithms. Alammary (2022) focuses on BERT models for Arabic text classification.

By presenting a comparative analysis, our systematic review contributes to the existing body of knowledge and provides researchers with a comprehensive understanding of the

advantages and limitations of previous reviews. This information aids in identifying research gaps and potential areas for further investigation within the field of Arabic language processing.

**Table 12: Comparative analysis between this systematic review and previous reviews**

| Source                      | Advantages                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Disadvantages                                                                                                                                                                                                                                                                                                        |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (Alwaneen et al. 2022)      | The review presents an in-depth examination of 25 contemporary Arabic Question Answering Systems (QASs) and expands upon previous surveys by categorizing recent studies based on the types of questions addressed. Additionally, it investigates the three components of QASs, the instruments employed by researchers, accessible Arabic QA datasets, evaluation metrics, and prospective research trajectories.                                                                                                                                                                                                                                    | <ul style="list-style-type: none"> <li>• Involves only QASs studies.</li> <li>• Covers only 25 research studies.</li> </ul>                                                                                                                                                                                          |
| (Alhumoud & Al Wazrah 2022) | The review examines the progression of research methodologies and subject matter in the field, employing keyword co-occurrence analysis in conjunction with a community detection algorithm. This investigation not only scrutinizes the interrelations between research methodologies and topics over the past twenty years, but also reveals prevailing hotspots and temporal trends, thereby offering valuable guidance for researchers. Additionally, the study furnishes comprehensive practical insights into the methods and topics of sentiment analysis, delineating technical trajectories, constraints, and prospects for future research. | <ul style="list-style-type: none"> <li>• Focuses on sentiment analysis.</li> <li>• Involves studies published in the Web of Science database, potentially excluding relevant papers in other databases.</li> <li>• Focuses on high-frequency keywords, potentially ignoring important low-frequency ones.</li> </ul> |
| (Al-Saleh & Menai 2016)     | The review offers a comprehensive review of Arabic text summarization by scrutinizing the extant literature in the domain. The analysis delves into diverse techniques for condensing Arabic text and appraising the results while emphasizing pertinent corpora, conferences, and workshops. Furthermore, the investigation probes the principal challenges encountered within the field.                                                                                                                                                                                                                                                            | <ul style="list-style-type: none"> <li>• Focuses solely on text summarization.</li> <li>• Covers only the methods of summarization and the corpora used.</li> <li>• It does not explore the processing or techniques employed in text summarization, nor does it survey the various tasks or</li> </ul>              |

|                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | applications of text summarization.                                                                                                                                                                                                                                                                                                                                                                               |
| (Alqudsi, Omar & Shaker 2014)        | The review concentrates on machine translation methodologies as delineated in prior research. It conducts an in-depth analysis of the linguistic characteristics of the Arabic language pertinent to machine translation, along with potential obstacles that may emerge. Furthermore, the study comprehensively reviews the primary techniques employed in machine translation from Arabic to English, assessing their respective merits and drawbacks.                                                                                                                                                                                                    | <ul style="list-style-type: none"> <li>• Focuses solely on machine translation techniques.</li> <li>• It does not investigate the methodologies utilized in text summarization or examine the diverse tasks or applications associated with machine translation.</li> <li>• It does not discuss whether the proposed solution in the reviewed papers is novel or a combination of existing approaches.</li> </ul> |
| (Alabbas, Al-Khateeb & Mansour 2016) | The review aims to identify and summarize the key attributes of various techniques employed in classifying Arabic text. Additionally, the review seeks to pinpoint potential areas for future research and emphasize the scarcity of standardized corpora in Arabic text. The review scrutinized 48 primary studies selected from 1464 papers, revealing that most research concentrates on Modern Standard Arabic while largely neglecting Colloquial Arabic, despite its widespread presence on social media platforms such as Twitter. Furthermore, the review employed empirical evidence to assess the accuracy of distinct classification algorithms. | <ul style="list-style-type: none"> <li>• The review mainly focuses on the processing phases of topics.</li> <li>• It does not cover the associated topics, applications, or tasks.</li> </ul>                                                                                                                                                                                                                     |
| (Farghaly & Shaalan 2009)            | The review examines several aspects of the Arabic language that present difficulties for ANLP applications. These encompass the sociolinguistic phenomenon of Arabic diglossia, the characteristics of the Arabic script, and particular linguistic features such as Arabic morphology, agglutination, and pro-drop. Furthermore, the review outlines solutions implemented by researchers to overcome these challenges. The absence of formal and explicit grammar for Modern Standard Arabic is also a hindrance to advancing more sophisticated ANLP systems.                                                                                            | <ul style="list-style-type: none"> <li>• It solely focuses on discussing the various aspects of the Arabic language that present challenges for ANLP applications.</li> <li>• It does not encompass the methodologies or techniques employed and neither addresses the processing phase nor the applications within the domain of ANLP.</li> </ul>                                                                |

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                      |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (Wahdan et al. 2020a) | <p>The review presents a comprehensive analysis of recent studies concerning text classification in the Arabic language, specifically focusing on the application of neural network algorithms. The research encompasses an array of neural network types, such as Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Feed-forward Neural Networks (FFNN), and Long Short-Term Memory (LSTM) networks, assessing their methodologies and accuracy in the classification of Arabic texts. Furthermore, the significance of evaluation criteria in attaining optimal classification accuracy is examined. The results indicated that the implementation of deep learning models has the potential to significantly advance text classification research within the context of the Arabic language.</p> | <ul style="list-style-type: none"> <li>• The review does not investigate alternative classification methodologies beyond neural networks, nor does it delve into the specific applications or subject areas where Arabic text classification is employed.</li> </ul> |
| (Alammary 2022)       | <p>BERT has emerged as a powerful instrument in NLP, owing to its capacity for context-based word prediction and the feasibility of pretraining it with accessible web-based textual data. While BERT has been adapted to accommodate a range of languages, including Arabic, its implementation in Arabic text classification remains constrained. This analysis endeavors to bridge this divide by synthesizing and contrasting diverse Arabic BERT models and assessing their performance in relation to English BERT models.</p>                                                                                                                                                                                                                                                                                    | <ul style="list-style-type: none"> <li>• The review is limited to surveying BERT models for Arabic text classification.</li> </ul>                                                                                                                                   |
| This review           | <p>This review offers a systematic review of 60 research articles, intending to deliver an in-depth understanding of the current state-of-the-art in ATC. The review organizes the gathered studies into three distinct categories: topic areas, tasks/applications, and ATC phases. Each category is further elucidated through its subcategories, encompassing all relevant topics, applications, methods, and techniques. Additionally, the review emphasizes the proposed solutions presented in each ATC study. This systematic review contributes to a holistic understanding of ATC's full Potential</p>                                                                                                                                                                                                         | <ul style="list-style-type: none"> <li>• The articles were retrieved from specific databases.</li> </ul>                                                                                                                                                             |

|  |                                                                |  |
|--|----------------------------------------------------------------|--|
|  | and furnishes valuable insights for future research endeavors. |  |
|--|----------------------------------------------------------------|--|

### 3.7. Systematic Review Discussion

In this systematic review, we categorized research publications on ATC into six primary domains: healthcare, legal, security and cybersecurity, social media, history, culture and religion, and agriculture. Our analysis revealed that social media content classification is the most prevalent research area in ATC, with a particular focus on Twitter. The findings indicated that the healthcare sector has received insufficient attention, thus highlighting the need for additional research in this domain. Within healthcare, we discovered that text classification has enhanced healthcare quality through various aspects, such as treatment, diagnosis, and management. Conversely, social media content may disseminate inaccurate information about healthcare issues, with over 50% of the information propagated on Twitter being false. Researchers in this field aim to raise awareness about the significance of disseminating accurate health-related information. In the legal sector, we identified only one study on ATC, which centered on the automatic classification of court rulings. On the other hand, the security and cybersecurity sector has seen extensive exploration in ATC, particularly in improving the monitoring of incidents and emergencies and mitigating cyber threats by identifying them in Instagram and Twitter posts. In the social media domain, our review showed that social media content is crucial for numerous individuals and applications and can be classified using ATC techniques. We underscored the importance of exercising caution with social media content due to the potential dissemination of misleading information. However, it is essential to note that our review did not uncover substantial research on ATC applications in specific sectors such as finance and transportation. This suggests opportunities

for further investigation and the development of ATC methodologies in these domains. ATC could enhance the predictability of air traffic operations within the transportation sector, thereby reducing delays and improving the overall passenger experience.

Moreover, we have comprehensively analyzed the gathered studies concerning ATC tasks and applications. We have categorized the extant research into nine primary task and application groups, examining each individually. This emphasizes the significance of ATC across various industries and the potential applications of ATC in tackling diverse challenges. For instance, gender identification, authorship identification, and dialect identification can be employed in law enforcement, whereas disease detection and awareness may aid mental health. Additionally, threat, spam, and incident detection can be utilized in emergencies and for monitoring the dissemination of false information and the propagation of infectious diseases. Ultimately, this underscores the potential of ATC in addressing these tasks and the necessity for continued research in this domain. Numerous areas warrant further investigation into ATC to enhance the comprehension of its capabilities in addressing extant challenges and limitations. Some of these areas encompass human factors. The impact of human factors on the utilization of ATC systems warrants exploration. This involves issues pertaining to training, workload, decision-making, and communication between controllers and pilots.

The ATC process comprises several stages: preprocessing, tokenization, and stemming. Most existing ATC research has concentrated on refining approaches and classifiers, while less attention has been devoted to developing new corpora and enhancing the preprocessing stage. However, recent studies have demonstrated that enlarging the dataset size significantly impacts ATC performance. Establishing a benchmark dataset for ATC is crucial

for comparing various ATC models. The issue of unbalanced datasets poses a considerable challenge in text classification. Several solutions have been proposed to address this problem, including the Class-Balanced (CB) SoftMax loss, under-sampling using Random Under-Sampling (RUS), and over-sampling using Random Over-Sampling (ROS). Stemming is regarded as a vital step in preprocessing and data cleansing. Numerous studies have investigated the influence of stemming on ATC, revealing its advantages, such as reduced corpus size, increased storage efficiency, and improved processing time. A critical limitation in the existing literature is the lack of an Arabic benchmark dataset for ATC model comparison. The unbalanced dataset issue is another significant limitation encountered by many studies. Although several solutions have been proposed, further research is necessary. Additionally, most studies have focused on refining approaches and classifiers, paying less attention to developing new corpora and enhancing the preprocessing stage. Consequently, there is a need for more research emphasizing these aspects. Furthermore, most studies have concentrated on news articles, leaving other text genres insufficiently explored.

There are several methodologies within the domain of ATC that merit further investigation. One such methodology is BERT, a state-of-the-art model that has demonstrated cutting-edge performance in natural language processing (NLP). The use of ensemble models, which combine BERT with neural networks, holds the potential for achieving improved outcomes. Additionally, the inherent benefits of BERT itself make it a valuable avenue for exploration in ATC.

Moreover, it is crucial to explore the development of ensemble models that incorporate unsupervised techniques prior to incorporating BERT. This approach has the potential to

generate promising results and should be thoroughly examined. Furthermore, novel preprocessing strategies can also be explored. For instance, instead of eliminating emojis, substituting them with their corresponding words could prove to be an effective preprocessing step.

Lastly, there is a need to investigate ensemble models that combine BERT with deep neural networks. Surprisingly, there is a lack of studies that have explored this combination, making it an area that holds potential for significant advancements in ATC.

Overall, the Systematic Review highlights the need for further investigation into BERT, ensemble models, unsupervised techniques, preprocessing strategies, and the combination of BERT with deep neural networks. These areas of exploration have the potential to contribute to advancements in Arabic text classification.

### **3.8. Systematic Review Summary**

In conclusion, this chapter provided an overview of text classification and its importance in organizing and categorizing documents based on their content. We specifically focused on Arabic text classification, highlighting the unique challenges it poses due to the complexity of the Arabic language.

We explored various techniques for text classification, including manual and statistical methods, as well as machine learning approaches. Manual text classification involves the creation of specific queries and categorizing texts based on predefined criteria, while statistical techniques rely on mathematical rules and principles. Machine learning techniques, such as supervised, unsupervised, and semi-supervised learning, have emerged as powerful tools for classifying unstructured texts. Neural networks, including feedforward neural networks,

convolutional neural networks, recurrent neural networks, and long short-term memory networks, have shown promising results in text classification tasks.

Furthermore, we discussed recent research on hate speech and offensive language identification in Arabic texts. Researchers have employed various approaches, such as pretrained models, preprocessing techniques, feature extraction methods, and ensemble methods, to enhance performance and address challenges like class imbalance and dialect normalization. The utilization of multiple pretrained models has been a common practice in these studies to capture contextual information effectively.

Overall, this chapter sheds light on the significance of text classification, particularly in the context of Arabic language processing. The advancements in machine learning and neural network techniques offer promising avenues for improving the accuracy and reliability of Arabic text classification, particularly in identifying offensive language and hate speech. The literature review also demonstrates the diversity of approaches and the continuous efforts in developing effective solutions for Arabic text classification tasks.

## Chapter 4: Methodology

In this chapter, we delve into the heart of our study, where we present the meticulously crafted methodology employed for the task of offensive language identification in Arabic text. The effective identification and mitigation of offensive language are crucial in today's digital age, where online communication plays a pivotal role in shaping public discourse. This chapter unfolds in two distinct sections, each vital to the successful execution of our research endeavor.

First, we start with a detailed exploration of the datasets used in our study. One of the significant challenges in offensive language identification research is the scarcity of standardized Arabic benchmark datasets specifically tailored for this purpose. We address this issue by meticulously selecting two datasets: the OSACT dataset and the SemEval dataset. These datasets serve as the cornerstone of our research, offering diverse and credible sources of Arabic text data enriched with offensive language expressions.

Next, we unveil the framework designed to tackle the task of offensive language identification in Arabic. This framework, the core of our research, is meticulously crafted to maximize the effectiveness of combining pretrained language models with deep learning techniques. We provide an in-depth exposition of this framework, elucidating its various components and their roles in processing Arabic text data.

In essence, this chapter is the blueprint that guides our journey through the realm of offensive language identification in Arabic text. By the end of this chapter, readers will have a comprehensive understanding of the methodology underpinning our research and the rationale

behind each carefully chosen element. It is within this methodology that we lay the foundation for robust and effective offensive language identification in the Arabic language, addressing the pressing need for advanced tools and techniques in this domain.

#### **4.1. Dataset Description**

One significant gap that has been addressed in the existing literature is the lack of a standardized Arabic benchmark dataset that can serve as a reliable and consistent means for evaluating and comparing Arabic Text Classification models. This research gap highlights the absence of a widely accepted and standardized dataset specifically designed for Arabic Text Classification. While various datasets may exist for this purpose, the lack of a unified benchmark dataset poses challenges in comparing the performance of different classification models across studies. The absence of a standardized dataset restricts the ability to make meaningful comparisons, impedes the advancement of Arabic Text Classification research, and hinders the identification of the most effective models and techniques for this specific domain. (Wahdan, Al-Emran & Shaalan 2023).

The chosen dataset was selected over other alternatives due to several justifications that make it the most suitable and relevant for the research purpose. Firstly, the OSACT dataset, introduced by Mubarak, Hassan & Chowdhury (2022b), has recently been utilized in related studies focusing on offensive text classification on Twitter. This allows for a direct performance comparison with existing research, enabling us to assess the proposed model's effectiveness in a consistent context.

Secondly, the SEMEval dataset, introduced by (Mubarak et al. 2021), has been widely used in the Multilingual Offensive Language Identification shared task (OffensEval 2020).

The popularity and adoption of this dataset in a renowned competition indicate its credibility and quality, making it a reliable choice for evaluation purposes.

Furthermore, both datasets contain Arabic text samples, which aligns perfectly with the research's focus on addressing offensive language in Arabic text. Utilizing datasets with Arabic content ensures that the proposed model is specifically tailored to the unique challenges and characteristics of the Arabic language, enhancing its relevance and applicability.

Additionally, the datasets provide a diverse set of offensive language expressions, encompassing various forms and contexts. This diversity allows for a comprehensive evaluation of the proposed model's ability to handle different types of offensive content, making it an asset for understanding the model's generalization capabilities in real-world scenarios.

By considering these factors and selecting OSACT and SEMEval datasets, the research aims to establish a strong foundation for evaluating the proposed model's performance and effectiveness in detecting offensive language in Arabic text. The justification for dataset selection is crucial in reinforcing the credibility and validity of the research findings and aligning the study with existing benchmarks and competitions in the field.

#### **4.1.1. OSACT Dataset**

The OSACT dataset offers a valuable resource for studying offensive content classification in Arabic text. Its wide coverage of offensive language categories and comprehensive annotation make it suitable for evaluating and comparing Arabic text

classification (ATC) models. Despite the imbalance among the categories, we have addressed this challenge through proper dataset partitioning.

The OSACT dataset comprises 12.7k tweets containing offensive content in both Modern Standard Arabic (MSA) and various Arabic dialects (DA). The dataset has been manually annotated for fine-grained abusive/offensive language categories, including hate speech, vulgar language, and violent content. In our research, we focused on the offensive and non-offensive categories. It is worth noting that this dataset is the most extensive categorized collection of Arabic tweets, covering a wide range of subjects, types, and dialects (Mubarak, Hassan & Chowdhury 2022b). Each tweet in the dataset has been evaluated by three annotators through crowdsourcing. However, one limitation of this dataset is the significant imbalance among the categories. To overcome this, we partitioned the dataset into three subsets for training, development, and testing, each comprising 70%, 10%, and 20% of the data. Table 13 illustrates the distribution and ratios of each labeled category.

**Table 13: Dataset statistics (Mubarak, Hassan & Chowdhury 2022b)**

| Class                 | Number of tweets | Percentage |
|-----------------------|------------------|------------|
| Offensive             | 4463             | 35%        |
| Not-Offensive (clean) | 8235             | 65%        |

#### **4.1.2. SemEval**

10,000 tweets were collected in Arabic using the Twitter API between April and May 2020. With the language filter set to “Arabic: lang:ar,” only tweets in Arabic were collected during the operation. Only tweets with two or more vocative particles (yA in Arabic) were considered for annotation to increase the possibility of including offensive content. Vocative

particles are frequently employed to direct discourse towards specific people or groups in offensive messages in various Arabic dialects. Due to this, 20% of the tweets in the final dataset were judged to be objectionable. Only Level A was annotated, and the annotation method involves hand labelling by a native speaker fluent in several dialects of Arabic. A random subset of offensive and non-offensive tweets underwent double annotation in order to measure inter-annotator agreement, and the results produced a Fleiss IAA score of 0.92. (Mubarak et al. 2021)

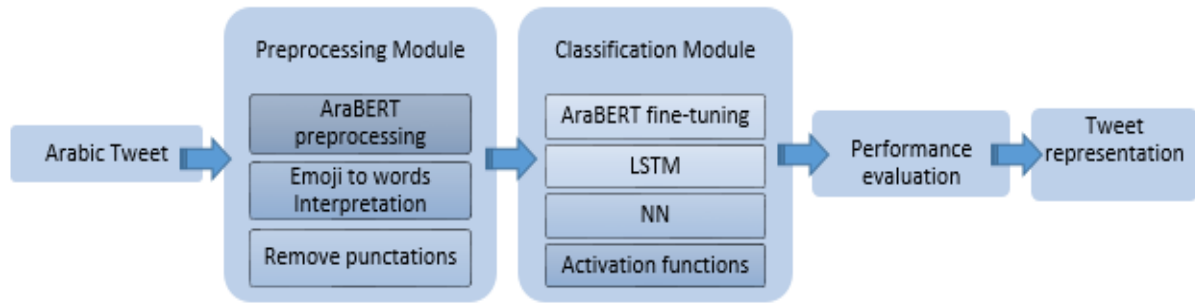
Table 14 illustrates the composition of each set, indicating the presence of both positive and negative tweets.

**Table 14: Dataset statistics (Mubarak et al. 2021)**

| Class                 | Number of tweets | Percentage |
|-----------------------|------------------|------------|
| Offensive             | 1982             | 20%        |
| Not-Offensive (clean) | 8018             | 80%        |

## 4.2. The Framework

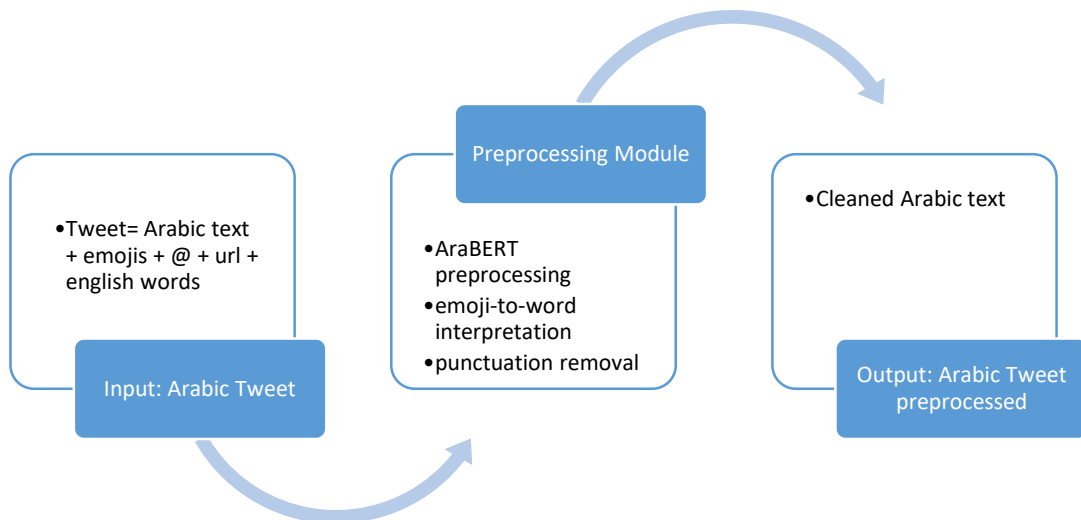
In this study, we specifically focus on investigating and evaluating the effectiveness of combining pretrained models with neural networks and deep learning for offensive language identification in Arabic. We aim to contribute to the existing literature by providing insights into the benefits and challenges of this approach and determining its impact on performance. The proposed framework is illustrated in Figure 17, comprising two primary modules: the Preprocessing Module and the Classification Module. In the following sections, we provide detailed explanations of each module.



**Figure 17: Proposed framework**

#### 4.2.1. Preprocessing Module

The Preprocessing Module aims to prepare the text data for the subsequent classification task. The Preprocessing Module comprises three submodules, namely AraBERT preprocessing, emoji-to-word interpretation, and punctuation removal, as depicted in Figure 18.



**Figure 18: Proposed preprocessing submodule**

#### 4.2.1.1. AraBERT Preprocessing

By performing these preprocessing steps, AraBERT preprocessing can improve the accuracy and performance of machine learning and NLP models that are trained on Arabic text data (Antoun, Baly & Hajj 2020) .

AraBERT preprocessing plays a crucial role in handling the unique characteristics of the Arabic language for natural language processing (NLP) tasks. The following are key aspects that can be further explained to provide a comprehensive understanding.

In Text Cleaning, Arabic text often contains unwanted characters, digits, and punctuations that may interfere with subsequent analysis. AraBERT preprocessing involves removing these non-essential elements to ensure a cleaner and more streamlined text data.

Example:

Original Text: "!!اليوم هو ١٤ سبتمبر، الطقس جميل"

Cleaned Text: "اليوم هو سبتمبر، الطقس جميل"

In Text Normalization, Arabic text may exhibit variations in spellings, formats, or word orders that do not affect the underlying meaning. AraBERT preprocessing standardizes the text by addressing these variations, making it consistent and easier to process.

Example:

Original Text: "مرحباً! كيف حالكم؟"

Normalized Text: "مرحباً! كيف حالك؟"

In Tokenization, Arabic text needs to be broken down into smaller units (token), such as words or phrases, to facilitate further analysis. AraBERT preprocessing employs tokenization techniques specifically tailored for Arabic text, ensuring accurate segmentation and representation of the text.

Example:

Original Text: "أنا أحب القراءة"

Tokenized Text: ["أنا", "أحب", "القراءة"]

In diacritization, the focus is on Arabic words, many of which come adorned with diacritics—those tiny marks positioned above or below specific letters, serving as vital indicators of vowel sounds. These diacritics play a pivotal role in ensuring accurate pronunciation and disambiguation, especially in the Arabic language. Their significance becomes particularly apparent when considering that Arabic words, sharing the same spelling, can encompass entirely distinct meanings contingent on the diacritics employed. This linguistic complexity underscores the need for robust tools like AraBERT to navigate and decipher the intricacies of diacritization in Arabic text. AraBERT's preprocessing includes diacritization as an integral step, ensuring the accuracy and clarity of the text.

Example:

Original Text: "كُتِبَ"

Diacritized Text: "كُتِبَ" (with diacritics for accurate pronunciation)

By incorporating these preprocessing steps, AraBERT preprocessing enhances the quality and suitability of Arabic text data for machine learning and NLP tasks. It helps in

reducing noise, improving the consistency of the data, and capturing the nuances specific to the Arabic language. Ultimately, AraBERT preprocessing contributes to improved accuracy and performance of models trained on Arabic text data. (Antoun, Baly & Hajj 2020) .

#### **4.2.1.2. Unleashing the Power of Emojis: Emoji-to-word Interpretation Algorithm**

In a relatively short span of time, emojis have emerged as prominent elements in our day-to-day communications. They have been widely regarded as a universal language, transcending linguistic and internet platform barriers, enabling communication on a global scale. This unique characteristic of emojis allows them to convey messages and emotions without relying on specific languages or linguistic patterns. Consequently, their extralinguistic information holds the potential to identify inappropriate content, irrespective of language-dependent knowledge or preferred linguistic structures (Dürscheid, C. and Siever 2017).

The utilization of emojis also addresses challenges posed by non-standard spellings of offensive words. Emojis serve as an invaluable resource for identifying abusive language, prompting several studies to adopt an automated emojis-based approach for collecting tweets with a significant proportion of malicious content, irrespective of language (Mei 2019).

To facilitate this, Mubarak employs emojis as anchors to compile tweets and manually annotate them based on finely grained categories of abusive/offensive language, including hate speech, vulgarity, and violent content. This dataset is then utilized to explore various aspects, such as the temporal patterns of emojis usage, extraction of offensive words with dialectal spellings and morphological variations, identification of hate speech targets, analysis of linguistic content in vulgar and profane words, and identification of common patterns in violent tweets (Mubarak, Hassan & Chowdhury 2022a).

The significance of emojis in offensive language identification cannot be overstated, as they effectively convey emotions, attitudes, and intentions within text. Emojis provide valuable contextual information that enhances the accuracy and comprehension of offensive language identification systems. Neglecting emojis would result in a loss of essential cues for detecting offensive content. (Mei 2019)

Previous studies have acknowledged the importance of emojis in offensive language identification and have endeavored to leverage them directly in the classification process. However, utilizing emojis in their raw form without preprocessing or interpretation has yielded unsatisfactory results (Mubarak, Hassan & Chowdhury 2022a).

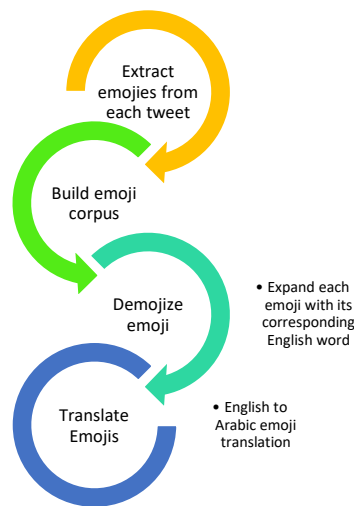
Therefore, we explore the novel approach of translating emojis into their corresponding literal meanings as a preprocessing step.

In offensive language identification, the effective utilization of emojis relies on their contextual information. To leverage this potential, we introduce the emoji-to-word interpretation algorithm as an innovative preprocessing technique. Instead of discarding emojis, this algorithm substitutes them with their literal meanings, preserving their informative value within the text.

#### **4.2.1.3      Emoji-to-word Interpretation Algorithm**

Emoji-to-word interpretation algorithm is a novel preprocessing approach that has not been introduced in Arabic text classification research, where emojis are substituted with their literal meaning instead of removing them entirely during the preprocessing step. It shows promise in improving offensive language identification in Arabic. According to (Mubarak, Hassan & Chowdhury 2022a), emojis were considered anchors in the OSACT dataset. In NLP,

anchors refer to fixed reference points used by the machine learning model to identify relevant patterns and features in the data. In this context, emojis were treated as anchors that contribute to the building of the dataset itself; that is, the presence or absence of emojis was a key feature in constructing the dataset and training the machine learning model. Therefore, the presence or absence of emojis interpretation was an important factor that we consider in the text data classification and analysis.



**Figure 19: Step-by-step process of emoji-to-word interpretation**

Therefore, we decided not to remove emojis from the dataset but rather interpret them with their corresponding meanings. However, we found no existing libraries nor algorithms capable of directly translating emojis into Arabic words, unlike the “demojize” function available for English. The 'demojize' function is a commonly used operation in natural language processing (NLP) and text analysis. Its primary purpose is to convert emojis, which are visual symbols often used in text-based communication, into their corresponding textual representations. In other words, it transforms emojis into plain text descriptions that convey the same meaning as the original visual symbols.

For example, the emoji 😊 might be converted into the textual representation ‘:smiling\_face\_with\_smiling\_eyes:’ This conversion allows NLP models and algorithms to work with emojis as if they were regular words, making it easier to analyze and process text data that includes emojis.

In the context of the Emoji-to-word interpretation algorithm, we use the 'demojize' function to translate emojis found in Arabic text into their English textual equivalents. This step is crucial because it bridges the gap between visual symbols and language, enabling us to work with emojis as part of the text analysis process. It ensures that the emojis' meanings are preserved and can be understood by NLP algorithms, enhancing the accuracy and comprehensibility of our offensive language identification system.

To overcome this limitation, we developed an algorithm that first collects all emojis from the dataset and then uses the ‘demojize’ function to translate them into English. We subsequently applied the translate function to convert the English words into Arabic, which we then saved in a dictionary. A detailed step-by-step of the Emoji-to-word interpretation process is presented in Figure 20.

A brief explanation of the main steps of the Emoji-to-word interpretation algorithm in plain language before presenting the pseudocode. The algorithm takes as input the total number of tweets (n) and the total number of emojis (j) in the document. It also takes the document itself, which is a collection of tweets.

### **Step 1: Initialization**

- Begin by initializing key counters: **x** for tweets and **y** for emojis.

- Set **x** to 1 (current tweet index) and **y** to 1 (current emoji index).

### **Step 2: Loop Through Tweets**

- Create a loop that sequentially processes each tweet in your dataset.
- For each tweet:

### **Step 3: Extract Emojis from Tweet**

- Inside the tweet loop, locate and collect all emojis found in the current tweet.
- Keep a record of these emojis.

### **Step 4: Move to the Next Tweet**

- After processing all emojis in the current tweet, increment the tweet index **x** by 1.
- This action moves to the next tweet in the dataset.

### **Step 5: Loop Through Emojis**

- Once all tweets have been examined, initiate another loop.
- This loop will traverse through the collected emojis one by one.
- For each emoji:

### **Step 6: Translate Emoji to Word**

- Retrieve the emoji at the current emoji index **y**.
- Translate this emoji into English words using the "demojize" function.

### **Step 7: Translate English to Arabic**

- Convert the English words obtained from the emoji into their corresponding Arabic meanings.

#### **Step 8: Move to the Next Emoji**

- After translating one emoji, increment the emoji index **y** by 1.
- This action progresses to the next emoji in the list.

#### **Step 9: Finish**

- Once all emojis have been processed, you will have a list of Arabic words that represent the emojis.
- This list serves as your "dictionary" for interpreting emojis in Arabic text.

*N: number of tweets*

*j: number of emojis*

*x: current tweet index*

*y: current emoji index*

*Document: Collection of tweets*

*Initialize  $x = 1$*

*Initialize  $y = 1$*

*While  $x \leq n$ , do:*

*Extract all emojis from tweet  $x$*

*Increment  $x$  by 1*

*End*

```

While y <= j, do:

    EmojiWord = Demojiz(y) # get the emoji at index y

    Translate EmojiWord to Arabic (find the corresponding Arabic meaning of EmojiWord)

    Increment y by 1

End

```

**Figure 20: Proposed emoji-to-word replacement algorithm pseudo code**

The resulting dictionary from this interpretation algorithm, which contains Arabic translations of emojis, is presented in appendix A and sample from it in Table 15. The dictionary was generated using the algorithm we developed.

**Table 15: Part from emojis dictionary we have created**

| #   | emoji | emoji_name                 | emoji_name_ar         |
|-----|-------|----------------------------|-----------------------|
| 0   | 😊     | :winking_face:             | : الوجه الغطس:        |
| 1   | 💧     | :pear:                     | : كَمَثْرَى:          |
| 2   | 💩     | :pile_of_poo:              | : كومة من البراز:     |
| 3   | 📺     | :wrapped_gift:             | : هدية ملفوفة:        |
| 4   | 🐝     | :honeybee:                 | : عسل النحل:          |
| ... | ...   | ...                        | ...                   |
| 572 | ❄️    | :snowflake:                | : ندفة الثلج:         |
| 573 | 🏠     | :name_badge:               | : اسم شارة:           |
| 574 | 🌲     | :evergreen_tree:           | : شجرة الخضرة:        |
| 575 | 🏡     | :house_with_garden:        | : منزل مع حديقة:      |
| 576 | ☔     | :umbrella_with_rain_drops: | : مظلة مع قطرات المطر |

#### 4.2.1.4. Punctuation Removal

Punctuation removal is an essential step in text preprocessing, including the AraBERT preprocessing. However, in certain cases, as observed in Table 16, we found that some punctuation marks were still present after the emoji interpretation step. To ensure the complete removal of punctuation marks and maintain consistency in the preprocessing pipeline, we repeated the process.

The removal of punctuation marks aims to eliminate characters such as commas, periods, exclamation marks, question marks, and other symbols from the text. These punctuation marks often do not contribute significantly to the semantic meaning of the text and can introduce noise or interfere with subsequent analysis.

By removing punctuation marks, we can improve the accuracy and effectiveness of language models and classification algorithms. It enables better identification of word boundaries, enhances the tokenization process, and facilitates the focus on the essential textual content.

Table 16 presents a selection of samples from the dataset, showcasing the impact of punctuation removal during the AraBERT preprocessing steps. Each row in the table includes the original tweet, followed by the tweet after undergoing AraBERT preprocessing, and finally, the tweet after interpreting emojis with their corresponding words and removing punctuations. This table serves as an illustrative example, demonstrating how the preprocessing steps, including punctuation removal, affect the tweets in the dataset.

The punctuation removal step is crucial for producing cleaner and more standardized text data. By eliminating unnecessary punctuation marks, we enhance the quality and

cohesiveness of the text, making it easier for subsequent analysis and classification tasks. Additionally, the removal of punctuation aids in improving the generalization ability of models by reducing the noise and distractions caused by these characters.

Overall, the inclusion of punctuation removal within the AraBERT preprocessing workflow ensures a more accurate representation of the underlying textual information, leading to improved performance in downstream tasks such as sentiment analysis, text classification, and information extraction.

**Table 16: Comparison of sample tweets before and after preprocessing**

| Original Tweets                                       | Tweets after AraBERT preprocessing                                                                                 | Tweet after replacing emoji and punctuation removal | Label<br>Not /<br>Offensive |
|-------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|-----------------------------|
| @USER ردينا ع التطنز 🤔                                | 🤔 [مستخدم] ردينا ع التطنز 🤔                                                                                        | ردينا التطنز الوجه المبتسم<br>القبضة القادمة        | OFF                         |
| وصارت فطاير البقالات غداء صحي 📄 URL                   | وصارت فطاير البقالات غداء صحي 📄 URL                                                                                | وصارت فطاير البقالات غداء<br>صحي استهجن لون ال      | NOT_OFF                     |
| @USER روي ليريده تلقين اشباه<br>...كثير بس ماحد زيك   | [مستخدم] روي ليريده تلقين اشباه<br>كثير بس ماحد زيكم مشفوح يخفق<br>ويطالعهم بكل انحطاط ويصورهم<br>🤔 الله يفشلكم بس | روي ليريده تلقين اشباه كثير<br>ماحد زيكم مشفوح.     | OFF                         |
| @USER مش باين حاجه خالص<br>🤔 <LF> عارف بقي 🤔          | [مستخدم] مش باين حاجه خالص<br>🤔 مش عارف بقي 🤔                                                                      | مش باين حاجه خالص الوجه<br>المثابر مش عارف بقي      | NOT_OFF                     |
| 👉 <LF><LF> اليوم_ الاثنين#<br>من المسلمين يقولك : 90% | 👉 الاثنين _ اليوم #<br>90 من المسلمين                                                                              | اليوم الاثنين التصفيق يقولك %<br>90 المسلمين        | NOT_OFF                     |

### **4.3. Classification Ensemble Module**

This section describes the proposed architecture for the classification module and its submodules along with the reasons for choosing them.

#### **4.3.1. AraBERT Pretraining**

AraBERT is an Arabic pretrained language model based on Google’s BERT architecture and employs the same BERT-Base configuration which was sourced from (Antoun, Baly & Hajj 2020). The model is pretrained and utilizes two large Arabic corpora that are publicly available. The first one is called the 1.5 billion words Arabic Corpus, which is a modern corpus consisting of over 5 million articles taken from ten major news sources across eight countries. The second one is OSIAN, which stands for Open-Source International Arabic News Corpus. This corpus includes approximately 3.5 million articles (equivalent to about 1 billion tokens) from 31 news sources located in 24 Arab countries. After eliminating duplicate sentences, the pre-training dataset’s final size was 70 million sentences, comprising roughly 24GB of text. This dataset encompasses news from various media across different Arab regions, making it a representative sample of a wide range of topics discussed in the Arab world. It is important to note that they kept words that include Latin characters because named entities, scientific or technical terms are frequently mentioned in their original language; hence, avoid losing valuable information. Following the pre-training, the model was fine-tuned on three different tasks: Sequence Classification, Named Entity Recognition, and Question Answering. (Antoun, Baly & Hajj 2020).

The results of the study conducted by Abu-farha & Magdy (2020) indicated that even after finetuning, the model performed better than other models and achieved state-of-the-art performance on Arabic dialects. This is the reason why we have chosen AraBERT to be the base of our classification module. According to Table 1, AraBERT, employed by Alzu'bi et al., (2022), achieved the highest performance. For our base model, we utilize AraBERT v2 (Antoun, Baly & Hajj 2020), which is publicly available in the Hugging Face library. We have fine-tuned AraBERT on a smaller target dataset, i.e., OSACT, to improve performance on the text classification task. Finetuning involved adjusting the model's hyperparameters.

#### **4.3.2. Enhancements by Adding LSTM**

In our proposed model, we enhance the finetuned AraBERT model by incorporating an LSTM (Long Short-Term Memory) layer and two linear layers.

The LSTM layer is added after the base model to capture sequential information in the input sequence. This layer is a type of RNN layer that can retain long-term dependencies in the input data, making it well-suited for processing text data. LSTM has consistently demonstrated high efficiency in ATC studies, as identified in a previous systematic review for a round 12 research conducted in 2020 (Wahdan et al. 2020b). It addressed the question, “Which type of neural network algorithm proved most efficient?” and the findings indicate that LSTM is the best and CNN comes next. The hidden state and cell state vectors in the LSTM layer are both set to 32, and the input size is set to the same as the output size of the previous linear layers. The number of recurrent layers in the network is set to 1.

### 4.3.3. Linear Layers

The output of the LSTM layer is passed through three linear layers of neural networks in order to improve the accuracy and efficiency of Arabic text classification using deep learning (Wahdan, Salloum & Shaalan 2021). We designed the three linear layers as fully connected neural network layers. They take the output of the LSTM layer and apply a linear transformation to it, followed by an activation function, to produce the output of the current layer. The “1st Linear layer” takes the output from the LSTM and reduces the number of features into 128, which are then fed into the “2nd Linear layer”. This layer reduces the number of features into 32, which are then fed into the “3rd Linear layer”. This layer reduces the number of features into 2, which is the number of classes in the task for classification of whether the tweet is offensive. The multiple layers allow the model to map the output of the pre-trained model to the required output for the analysis task.

The first linear layer maps the output of the LSTM layer to a hidden space of size 128. It applies a linear transformation to the input, followed by the “ReLU” activation function. Next, a “Dropout” operation is applied to introduce regularization and prevent overfitting. Dropout randomly sets a fraction of the input elements to zero during training.

The second linear layer maps the output of the first linear layer to a hidden space of size 32. It applies a linear transformation to the input, followed by the “ReLU” activation function. Next, a “Dropout” operation is applied again.

Finally, the third linear layer takes the output of the previous layer, applies a linear transformation, and maps it to the final output space of size 2. This represents the classification probabilities for the two intended classes in the task. The output of the second

liner layer is passed through the “Sigmoid” activation function. The Sigmoid function squashes the input values between 0 and 1, providing the final classification probabilities for the two classes. It is commonly used for binary classification tasks, such as classifying whether tweets are offensive.

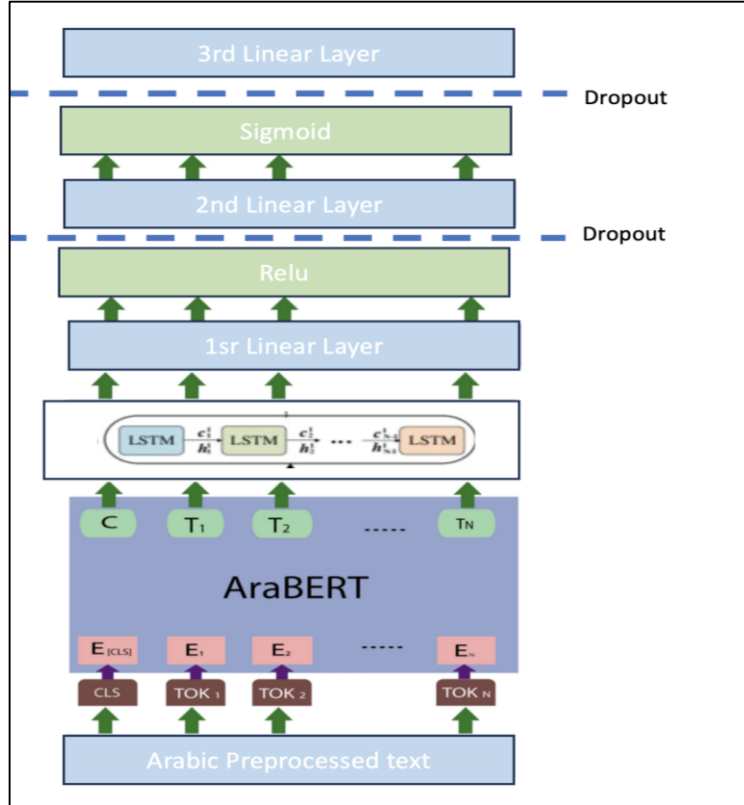
#### **4.4. Significance of Ensemble Approach in Combining AraBERT with the Custom Model**

The proposed model defines a custom PyTorch model for sequence classification. During training, the “CustomModel” takes input sequences and their corresponding labels, pass them through the pre-trained AraBERT model to obtain a sequence of hidden representations. This sequence is then fed into the LSTM layer, and subsequently through the linear layers and activation functions.

The Custom Model is designed to enhance the pre-trained transformer-based language model by incorporating additional layers that effectively capture sequential information in the input text. These added layers, along with the activation functions, enable the model to effectively map the output of the pre-trained model to the text classification task.

Furthermore, this model is designed to prevent overfitting and generate probabilities for the two classes using multiple approaches. By combining these design choices, the model exhibits good generalization on unseen data and provides reliable probabilities for the text classification task. The model is an ensemble of several sub-models, including AraBERT, Linear neural network, LSTM, and activation functions, as depicted in Figure 21. Overall, the suggested revisions aim to provide a clearer and more concise explanation of the proposed

architecture and its components. Arabic preprocessing is already explained in the previous section.



**Figure 21: Proposed architecture for the classification module**

The combination of AraBERT, a powerful pretrained language model, with the Custom Model architecture plays a pivotal role in enhancing the performance of offensive language identification in Arabic text.

AraBERT's pretraining on extensive Arabic corpora, including the 1.5 billion words Arabic Corpus and OSIAN, forms a strong foundation for understanding Arabic text, including dialects and various linguistic nuances.

The Custom Model, built on top of AraBERT, extends its capabilities by incorporating additional layers, such as an LSTM and linear layers. This augmentation enables the model to

capture sequential information, a crucial aspect of understanding the context in offensive language identification.

Ensemble these components, including AraBERT, Linear neural network, LSTM, and activation functions, offers a comprehensive approach to tackling overfitting while providing reliable probabilities for text classification.

The LSTM layer within the Custom Model is specifically designed to capture long-term dependencies in the input text, a feature essential for processing textual data effectively.

The linear layers further refine the model's output, reducing feature dimensions and applying activation functions to produce classification probabilities for offensive and non-offensive text.

The ensemble approach ensures that the model's generalization capabilities extend to unseen data, making it robust and suitable for real-world applications.

Overall, the combination of AraBERT and the Custom Model through an ensemble approach represents a well-rounded solution for offensive language identification in Arabic text, leveraging the strengths of both pretrained language models and deep learning techniques.

## **4.5. Conclusion**

In this chapter, we have presented a comprehensive methodology for offensive language identification in Arabic text. This methodology encompasses two key components: dataset description, and the framework.

The dataset description section addressed the critical issue of the lack of a standardized Arabic benchmark dataset for evaluating Arabic Text Classification models. To bridge this gap, we carefully selected two datasets: the OSACT dataset and the SemEval dataset. These datasets were chosen for their relevance, credibility, and coverage of offensive language expressions in Arabic text.

The OSACT dataset, with its extensive collection of tweets containing offensive content in both Modern Standard Arabic and various dialects, provided a rich source for evaluating offensive language identification models. Despite category imbalances, proper dataset partitioning was performed to ensure reliable training, development, and testing subsets. The SemEval dataset, with its 10,000 Arabic tweets collected through the Twitter API, offered an additional source for evaluating the proposed models. The inclusion of tweets with vocative particles helped increase the likelihood of capturing offensive content.

In the framework section, we outlined the proposed offensive language identification framework, which aimed to evaluate the effectiveness of the ensemble model combining pretrained models with neural networks and deep learning for this task. The framework consisted of two primary modules: The Preprocessing Module and the Classification Ensemble Module. The Preprocessing Module involved several steps, including AraBERT preprocessing, emoji-to-word interpretation, and punctuation removal, to prepare the text data for classification effectively. The novel emoji-to-word interpretation algorithm was introduced to harness the contextual information provided by emojis and improve model performance.

The Classification Ensemble Module incorporated a custom PyTorch model for sequence classification. This model extended the pretrained AraBERT model with additional

layers, including an LSTM layer and linear layers, to capture sequential information in the input text. The choice of AraBERT as the base model was grounded in its strong performance on Arabic text classification tasks, and fine-tuning on the OSACT dataset further enhanced its suitability for offensive language identification. The LSTM layer and linear layers were added to enable the model to effectively map the output of AraBERT to the binary classification task of identifying offensive and non-offensive text.

In conclusion, this chapter outlines a robust methodology for offensive language identification in Arabic text, addressing the need for standardized datasets and proposing a comprehensive framework that combines pretrained models with deep learning techniques. The carefully selected datasets, preprocessing steps, and model architecture lay the foundation for rigorous evaluation and comparative analysis in subsequent chapters. This research contributes to advancing the field of Arabic Text Classification and offers valuable insights into addressing offensive language in the Arabic language domain.

## **Chapter 5: Experiments and Results**

This chapter, we venture into the experimental setup, which is instrumental in evaluating the performance and efficacy of our proposed methodology. Our research seeks to answer pivotal questions regarding the effectiveness of this approach, and the experimental setup serves as the arena where these questions are put to the test.

Furthermore, the chapter presents the experimental results obtained from the conducted study. It describes the outcomes achieved. The experimental results showcase the performance of the proposed approach or model in terms of the identified metrics, highlighting its strengths and weaknesses.

Moreover, the chapter aims to compare the research results with previous related work in the field. It provides a critical analysis and discussion of how the proposed approach or model compares to existing methods or approaches used in similar studies. This comparison helps to identify the advancements, improvements, or limitations of the proposed research in relation to the state-of-the-art techniques.

Overall, this chapter presents the experimental results obtained, and conducts a comparative analysis with previous related work to establish the novelty and contribution of the research study.

## 5.1. Experimental Setup

### 5.1.1. Hardware Configuration

In the experimental setup for this study, the hardware configurations were deployed on Google Cloud Platform. The computational resources, including GPU instances (V100/T4), were provisioned through Google Cloud services. The use of cloud-based infrastructure facilitated scalability, flexibility, and ease of resource management during the model training process. Hardware configuration is clarified in table 17.

**Table 17: Hardware Configuration**

|              |          |
|--------------|----------|
| GPU Type     | V100/T4  |
| Maximum Size | 16GB GPU |

### 5.1.2 Software Configuration

To train the proposed model, we start by fine-tuning the parameters of AraBERT. For this purpose, we employ the AdamW optimizer, which is a variant of the Adam optimizer designed to incorporate weight decay regularization and mitigate overfitting. Additionally, we utilize a learning rate scheduler to gradually reduce the learning rate over time, facilitating a more gradual convergence of the model as depicted in table 18.

**Table 18: Software configuration**

| Model Architecture        | AraBERT                         |
|---------------------------|---------------------------------|
| Optimizer                 | AdamW                           |
| Learning Rate             | 2e-5                            |
| Adam Epsilon              | 1e-8                            |
| Mixed Precision Training  | Enabled (fp16) for V100/T4 GPUs |
| Cross-Validation Strategy | 5-fold                          |

The initial learning rate is set to 2e-5, and the Adam epsilon value is set to 1e-8. The model is trained for 2 epochs with a batch size of 16 per device, and the gradient accumulation steps are set to 2 to reduce the memory requirements. We also enable mixed precision training (fp16) when using V100 or T4 GPUs to speed up the training process. The maximum batch size is set to 64 on a 16GB GPU with a maximum sequence length of 128.

During training, we used a 5-fold cross-validation approach to evaluate the performance of our custom model (de Paula et al. 2022). We split the dataset into five equal parts and trained our model on four of them while using the remaining part as the validation set. We repeated this process five times so that each part of the dataset was used for validation exactly once. a checkpoint is saved for the corresponding subtask when its F1-macro score improves. We set the warm-up ratio to 0 and use the 'epoch' evaluation and save strategies. The total number of steps used for training is determined by the number of epochs, batch size, and number of gradient accumulation steps. We load the best model at the end based on the

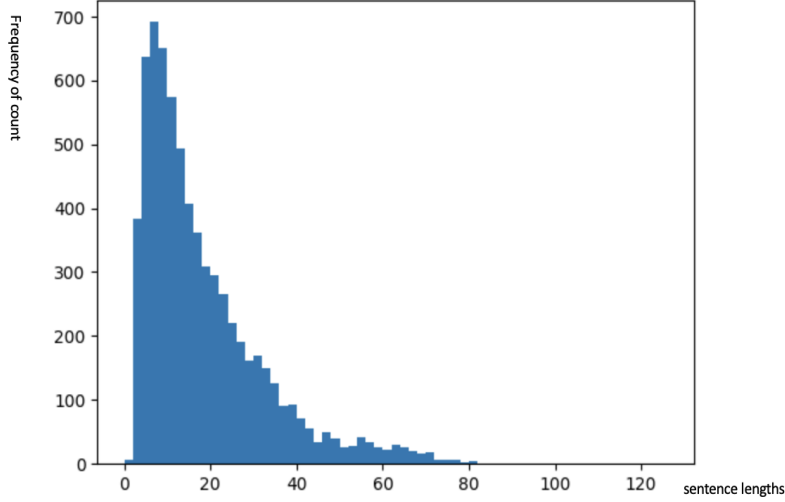
macro F1 score, and the “greater\_is\_better” flag is set to True. Finally, we use a fixed seed value of 25 for reproducibility.

We used the macro F1-score as the metric for selecting the best model because it balances the F1-scores of both classes. We also used early stopping to prevent overfitting and saved the best model based on the validation set performance. Finally, we used a tokenizer that is specific to the model architecture we used.

## **5.2. Experimental Results**

### **5.2.1. Preprocessing Results**

We need to check the tokenized sentence length to decide on the maximum sentence length value. The histogram in Figure 22 showing the distribution of sentence lengths in Osact dataset. The x-axis represents the range of sentence lengths, while the y-axis represents the frequency or count of sentences falling into each length range.



**Figure 22: Distribution of sentence lengths in OSACT dataset**

Table 19 presents the accuracy and f1 results of applying the preprocessing techniques. For the OSACT dataset, the models trained on the OSACT dataset with AraBERT preprocessing using the proposed model achieved the highest accuracy and F1-scores of 0.870 and 0.853, respectively. Additionally, the introduction of other preprocessing techniques, such as interpreting emojis with words or emoji embeddings, removing Arabic stop words, removing punctuations, and applying AraBERT preprocessing, resulted in improvements in accuracy and F1-score give accuracy ranges from 0.84 to 0.86 and f1score ranges from 0.82 to 0.84. Also. Specifically, the replacement of emojis with words demonstrated superior performance compared to the utilization of emoji embeddings, indicating promising prospects for future advancements in model development.

Furthermore, we evaluated the proposed model on the semEval dataset, where the models achieved higher accuracies ranging from 0.88 to 0.93 and F1-scores ranging from 0.87 to 0.90.

**Table 19:Preprocessing techniques performance metric results**

| Dataset                                           | Preprocessing<br>(All without balancing) |                                  |                                                                                           | Additional Classifier to<br>AraBERT |     |        |     | Activation<br>function |         | Performance metric |              |                |                | Confusion Matrix                      |
|---------------------------------------------------|------------------------------------------|----------------------------------|-------------------------------------------------------------------------------------------|-------------------------------------|-----|--------|-----|------------------------|---------|--------------------|--------------|----------------|----------------|---------------------------------------|
|                                                   | Interpret emoji<br>with words            | Map emoji to<br>emoji embeddings | Other<br>preprocessing:<br>stop words<br>removal,<br>punctuations<br>removal &<br>AraBERT | 2NN                                 | 3NN | 4 LSTM | CNN | ReLU                   | Sigmoid | Accuracy           | F1-score     | Recall         | Precision      |                                       |
| OSACT<br>dataset                                  | √                                        |                                  | √                                                                                         | √                                   |     |        | √   | √                      | √       | 0.840              | 0.820        | 0.830          | 0.827          | [1082<br>164]<br>[ 150<br>569]        |
| OSACT<br>dataset                                  | √                                        |                                  | √                                                                                         | √                                   |     | √      |     | √                      |         | 0.840              | 0.820        | 0.831          | 0.829          | [1080<br>162]<br>[ 152<br>569]        |
| <b>OSACT<br/>dataset</b>                          | √                                        |                                  | √                                                                                         | √                                   |     | √      |     | √                      | √       | <b>0.860</b>       | <b>0.840</b> | 0.842          | 0.850          | [785<br>78]<br>[ 90<br>313]           |
| <b>OSACT<br/>dataset<br/>(Proposed<br/>Model)</b> |                                          |                                  | AraBERT<br>preprocessing<br>only                                                          | √                                   |     | √      |     | √                      | √       | <b>0.870</b>       | <b>0.853</b> | <b>0.850</b>   | <b>0.840</b>   | <b>[771<br/>92]<br/>[ 77<br/>326]</b> |
| OSACT<br>dataset                                  |                                          | √                                | √                                                                                         |                                     | √   | √      |     | √                      | √       | 0.830              | 0.820        | 0.82           | 0.83           | [1479<br>175]<br>[ 226<br>661]        |
| OSACT<br>dataset                                  |                                          | √                                | √                                                                                         |                                     |     |        |     |                        |         | 0.810              | 0.770        | 0.800          | 0.760          | [763<br>110]<br>[106<br>286]          |
| OSACT<br>dataset                                  | √                                        |                                  | √                                                                                         |                                     | √   |        |     | √                      |         | 0.830              | 0.810        | 0.82           | 0.83           | [1479<br>175]<br>[ 226<br>661]        |
| semEval<br>Dataset                                |                                          | √                                | √                                                                                         |                                     | √   | √      |     | √                      | √       | 0.880              | 0.870        | 0.875<br>0.865 | 0.875<br>0.865 | [832<br>48]<br>[ 52<br>350]           |
| semEval<br>Dataset                                | √                                        |                                  | √                                                                                         | √                                   |     | √      |     | √                      | √       | <b>0.900</b>       | <b>0.890</b> | 0.895          | 0.885          | [845<br>25]<br>[ 43<br>359]           |
| semEval<br>Dataset<br><b>(Proposed<br/>Model)</b> |                                          |                                  | AraBERT<br>preprocessing<br>only                                                          | √                                   |     | √      |     | √                      | √       | <b>0.930</b>       | <b>0.900</b> | <b>0.910</b>   | <b>0.920</b>   | <b>[875<br/>5]<br/>[ 31<br/>371]</b>  |

### 5.2.2. Classification Results

Table 20 presents the performance metrics, accuracy, and F1-macro score, of various classifier models. The models employ different techniques such as Decision Tree, Multinomial NB, Linear SVC, Random Forest, Ali Safaya (Bert-mini-Arab) (Safaya, Abdullatif & Yuret 2020), Bertopic (Unsupervised) (Grootendorst 2022) with logistic regression, and different combinations of AraBERT. The baseline of applying AraBERT's Mubarak, Hassan & Chowdhury (2022a) achieved an accuracy of 0.82, serving as a baseline for comparison with other models. Models utilizing TFIDF with Decision Tree, Multinomial NB, Linear SVC, and Random Forest showed significantly lower accuracies ranging from 0.26 to 0.32. Ali Safaya (Bert-mini-Arab) (Safaya, Abdullatif & Yuret 2020) achieved an accuracy of 0.22, indicating poor performance in the given task.

Bertopic (Unsupervised) combined with logistic regression achieved a relatively high accuracy of 0.68. However, the other combinations and finetuned models did not yield significant performance improvements in this research, but they may be potential areas for future exploration. Ensemble models with a base of AraBERT, followed by various combinations of neural network architectures, such as 2NN, 3NN, and deep learning models, such as LSTM, and CNN, along with different activation functions (e.g., ReLU, Sigmoid), yielded improved accuracies ranging from 0.80 to 0.86. The proposed model, incorporating AraBERT preprocessing, 3 NN, 4 layers of LSTM, AraBERT, and ReLU and Sigmoid activation functions, achieved state-of-the-art performance with the highest accuracy of 0.870.

Additionally, experiments were conducted on the semEval dataset using the proposed model, resulting in an accuracy of 0.93 and an F1-score of 0.90, as indicated in Table 18.

Table 20 provides a comprehensive comparison between our proposed model and previous studies conducted on the same dataset, enabling a meaningful assessment of model performance. The results depicted in the table explicitly demonstrate that our model surpasses all other models, thereby establishing its superiority in terms of performance. In conclusion, the analysis of Table 20 and 22 unequivocally underscores the exceptional performance of our proposed model, which achieved an impressive accuracy of 0.87.

**Table 20: Classifier results**

| <b>Classifier Model</b>                                                             | <b>Accuracy</b> | <b>F1-macro</b> | <b>Recall</b> | <b>Precision</b> | <b>Confusion Matrix</b>  |
|-------------------------------------------------------------------------------------|-----------------|-----------------|---------------|------------------|--------------------------|
| <b>Baseline AraBERT (Mubarak, Hassan &amp; Chowdhury 2022b).</b>                    | 0.820           | 0.800           | -             | -                | -                        |
| <b>TFIDF + Decision Tree</b>                                                        | 0.320           | 0.300           | 0.320         | 0.280            | [462 943]<br>[685 274]   |
| <b>TFIDF + Multinomial NB</b>                                                       | 0.260           | 0.210           | 0.260         | 0.200            | [1176 229]<br>[ 278 499] |
| <b>TFIDF + Linear SVC</b>                                                           | 0.260           | 0.220           | 0.240         | 0.200            | [1035 370]<br>[ 400 430] |
| <b>TFIDF + Random Forest</b>                                                        | 0.260           | 0.240           | 0.260         | 0.230            | [787 618]<br>[ 220 610]  |
| <b>Ali safaya (Bert-mini-Arabic)</b>                                                | 0.220           | 0.210           | 0.220         | 0.200            | [1552 370]<br>[ 629 194] |
| <b>Bertopic (Unsupervised) + logistic regression</b>                                | 0.680           | 0.40            | 0.670         | 0.290            | [871 446]<br>[ 352 499]  |
| <b>AraBERT preprocessing<br/>AraBERT (5 epochs )</b>                                | 0.800           | 0.770           | 0.790         | 0.820            | [875 91]<br>[ 207 639]   |
| <b>AraBERT preprocessing<br/>All Dataset+ SMOTE + AraBERT</b>                       | 0.790           | 0.750           | 0.770         | 0.740            | [564 682]<br>[ 186 553]  |
| <b>AraBERT preprocessing<br/>2nn + ReLU and Sigmoid + LSTM (tuned)<br/>+AraBERT</b> | 0.850           | 0.830           | 0.860         | 0.820            | [924 169]<br>[ 157 569]  |
| <b>AraBERT preprocessing<br/>ReLU + 3nn + AraBERT</b>                               | 0.820           | 0.810           | 0.820         | 0.810            | [902 185]<br>[ 168 659]  |
| <b>AraBERT preprocessing<br/>ReLU and Sigmoid + 3nn + AraBERT</b>                   | 0.830           | 0.820           | 0.82          | 0.83             | [1479 17<br>[ 226 661]   |

|                                                                                                                                                                  |              |              |              |              |                               |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------------|--------------|--------------|-------------------------------|
| <b>AraBERT preprocessing<br/>3nn + AraBERT</b>                                                                                                                   | 0.810        | 0.800        | 0.810        | 0.800        | [1130 202]<br>[ 230 604]      |
| <b>AraBERT preprocessing<br/>ReLU and Sigmoid + 3nn + LSTM +<br/>AraBERT + Increased neurons</b>                                                                 | 0.860        | 0.840        | 0.842        | 0.850        | [785 78]<br>[ 90 313]         |
| <b>replace emoji with words.<br/>remove Arabic stop words.<br/>remove punctuations.<br/>AraBERT preprocessing<br/>2NN + CNN + ReLU + Sigmoid +<br/>AraBERT</b>   | 0.840        | 0.820        | 0.830        | 0.827        | [1082 164]<br>[ 150 569]      |
| <b>replace emoji with words.<br/>remove Arabic stopwords.<br/>remove punctuations.<br/>AraBERT preprocessing<br/>2NN + LSTM + ReLU and Sigmoid +<br/>AraBERT</b> | 0.860        | 0.840        | 0.843        | 0.849        | [785 78]<br>[ 90 313]         |
| <b>Proposed Model (AraBERT<br/>preprocessing+ 3 NN + 4 layers LSTM+<br/>AraBERT + ReLU and Sigmoid)</b>                                                          | <b>0.870</b> | <b>0.853</b> | <b>0.851</b> | <b>0.845</b> | <b>[771 92]<br/>[ 77 326]</b> |

Table 21 presents a sample comparison of predicted and actual classes for the test data. It is observed that most of the errors in prediction cannot be easily identified even by humans, except for example 5 where the prediction differs from the actual class.

**Table 21: Comparison of predicted and actual classes for test data (sample)**

|    | Text                                                                                                                                                                                                | Predicted Class | Actual Class |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|--------------|
| 1  | 🤪🖋️[مستخدم] افطرت عليك بقاء واثنين من فروخها الجن                                                                                                                                                   | OFF             | NOT_OFF      |
| 2  | 🙄داليا _ مبارك مادري ليش نقرت #                                                                                                                                                                     | OFF             | NOT_OFF      |
| 3  | مستخدم] ابديت السناپ الجديد ✕ حاس الناس حوس ، أشوف مشاهير . تضيفني ، مشاهير تتابع يومياتي ، [ RT<br>🤪 ابديت كرهني بالسناپ كله                                                                       | NOT_OFF         | NOT_OFF      |
| 4  | 🖋️🖋️💖💖[مستخدم] هيبه والله واليدد                                                                                                                                                                    | NOT_OFF         | NOT_OFF      |
| 5  | 🤪 URL اكيد اخس شي                                                                                                                                                                                   | OFF             | NOT_OFF      |
| 6  | 🙄مابي شي الحين غير فراشي                                                                                                                                                                            | NOT_OFF         | NOT_OFF      |
| 7  | 🤪[مستخدم] اقسام بالله عيب اللي يصير [ RT                                                                                                                                                            | NOT_OFF         | NOT_OFF      |
| 8  | 🤪🤪[مستخدم] اه هذا واش عندوا زيادة                                                                                                                                                                   | NOT_OFF         | NOT_OFF      |
| 9  | 🤪🕌[مستخدم] اذكر الله يامل الحتسه                                                                                                                                                                    | NOT_OFF         | NOT_OFF      |
| 10 | 🤪 URL ! القرد عاد نفسه فرد من العائله 🤪🖋️[مستخدم] ابتلشت بالقرد [ RT                                                                                                                                | OFF             | OFF          |
| 11 | 🤪[مستخدم] اما صاحبه                                                                                                                                                                                 | NOT_OFF         | NOT_OFF      |
| 12 | "كذب لا صدق ولا كذب . . لعب بالمشاعر فقط . وإذا تعلق في من تحب _ حب _ الانترنت _ نت _ صدق _ او #<br>💖🕌🕌هنا وقتها تحط موجوع قلبي 🤪 . . "حظرك                                                         | NOT_OFF         | NOT_OFF      |
| 13 | 🤪[مستخدم] حيوان هذا                                                                                                                                                                                 | OFF             | OFF          |
| 14 | امردغه أمردغه امردغه ورجاء 🖋️[مستخدم] # كلمه _ لحظك احتاج غرفة فاضيه فيها . . أنا وحظي ونعال [ RT<br>🤪🤪🤪🤪🤪🤪🖋️ لحد يتدخل                                                                             | NOT_OFF         | NOT_OFF      |
| 15 | # كله _ بيتعوض _ الا 🖋️🖋️الا الحاجه اللي مابتتعوض                                                                                                                                                   | NOT_OFF         | NOT_OFF      |
| 16 | [مستخدم] نبيغاك تفك النصر من شرك كافي مدهرته بتطبيلك وتضليلك لادارة الجهل السابقة التي حاست النصر واضعفته وانت كنت المضلل الكبير لها ! ! خلاص يا عبدالعزیز كرتك احترق والمفروض تتوارى عن النصر فليس | OFF             | OFF          |

|    |                                                                                                                                                                                                                   |             |              |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------------|
|    | 👉 لك قبول من الجميع !                                                                                                                                                                                             |             |              |
| 17 | ✍️ [مستخدم] انا مقرره اتوب بس ذا طلعتني عن طوري                                                                                                                                                                   | NOT_<br>OFF | NOT_<br>_OFF |
| 18 | الاشكال هذي اللي تطلع 📄 □ أبـ مدعـ الوايـ ربعـ مليون # سعوديـ الاصلـ و_ المنشأ #<br>وللاسف الاعلام السعودي نايم ومكير الوساده ومخليهم على راحتهم يمثلون الوطن 😊بالاعلام وتقول انا سعوديـ<br>📄 URL 📄 بشكل غير صحيح | OFF         | NOT_<br>_OFF |
| 19 | ✍️ هالتونسية تطلق علينا هالوقت معليش تمنون : اجل نهار ومساء                                                                                                                                                       | NOT_<br>OFF | NOT_<br>_OFF |
| 20 | 📄 [مستخدم] بنت الفراش ومنقهرة وعشان كذا نطيتي في منشني وتقولين انقلعي هه الخلا مئاك متخلفه                                                                                                                        | OFF         | OFF          |

**Table 22:Performance comparison of the proposed model with prior studies**

| Paper                               | Accuracy      | F1-score     |
|-------------------------------------|---------------|--------------|
| Baseline                            | <b>0.651</b>  | <b>0.394</b> |
| (Mubarak, Hassan & Chowdhury 2022b) | 0.820         | 0.800        |
|                                     | 0.840         | 0.823        |
| (AlKhamissi & Diab 2022)            | 0.860         | 0.845        |
| (Mostafa, Mohamed & Ashraf 2022)    | Not mentioned | 0.851        |
| (Nessir et al. 2022)                | 0.854         | 0.839        |
| (de Paula et al. 2022)              | 0.841         | 0.827        |
| (Makram et al. 2022)                | 0.800         | 0.780        |
| (Shapiro, Khalafallah & Torki 2022) | 0.856         | 0.841        |
| (Elkaref & Abu-Elkheir 2022)        | 0.765         | 0.742        |
| (Alzu'bi et al. 2022)               | 0.864         | 0.849        |
| <b>Proposed model</b>               | <b>0.870</b>  | <b>0.853</b> |

### 5.2.3. Incorporating additional dataset

In the context of Arabic text classification, incorporating additional relevant data has the potential to improve performance by enhancing the size and quality of the target dataset. This can be achieved through various means, such as training pre-trained models on similar

datasets or integrating external knowledge sources like word embeddings or knowledge graphs. However, it is essential to assess the impact of such data incorporation on the classification performance accurately.

In this study, we aimed to investigate the effect of incorporating additional data on the performance of the proposed model. We considered two distinct datasets: the OSACT dataset and the semEval dataset. Both datasets were subjected to preprocessing steps, including the removal of stop words, punctuation, and applying the AraBERT preprocessing.

Initially, we trained the proposed model using the individual datasets, without any data balancing techniques. The AraBERT preprocessing was employed exclusively for both datasets, ensuring consistent treatment of the input text. The model was evaluated using multiple activation functions, including ReLU, Sigmoid, and others, in conjunction with various classifiers such as 2NN, 3NN, different layers of LSTM, and CNN.

Table 23 presents the performance comparison of the proposed model on OSACT, semEval, and both Combined Datasets. The results revealed that when trained solely on the OSACT dataset, the proposed model achieved an accuracy of 0.870 and an F1-score of 0.853. Similarly, when trained exclusively on the semEval dataset, the model demonstrated improved performance with an accuracy of 0.930 and an F1-score of 0.900.

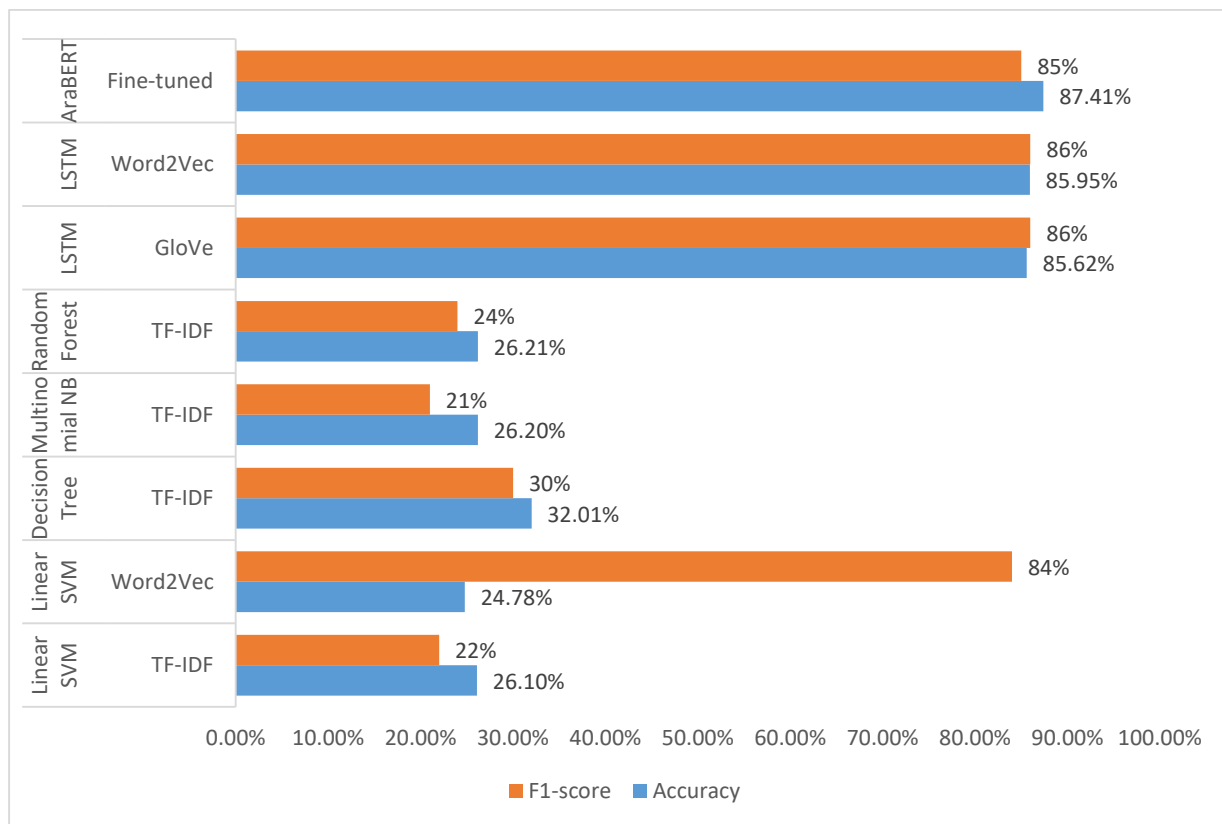
However, when we combined both the OSACT and semEval datasets, the performance of the model exhibited a decrease in accuracy and F1-score. The accuracy dropped to 0.830, while the F1-score declined to 0.820.

**Table 23: Performance comparison of the proposed model on OSACT, semEval, and combined datasets**

| Dataset                           | Accuracy | F1-score | Recall | Precision | Confusion                |
|-----------------------------------|----------|----------|--------|-----------|--------------------------|
| OSACT dataset                     | 0.870    | 0.853    | 0.851  | 0.845     | [771 92]<br>[ 77 326]    |
| semEval Dataset                   | 0.930    | 0.900    | 0.41   | 0.41      | [0 0]<br>[ 0 315]        |
| Combining OSACT + semEval Dataset | 0.830    | 0.820    | 0.82   | 0.83      | [1479 175]<br>[ 226 661] |

#### 5.2.4. Feature Extraction Techniques

Figure 23 presents the performance metrics of various classifiers and feature extraction techniques used for Arabic text classification. The metrics evaluated include accuracy and F1-score, which provide insights into the classification models' effectiveness and robustness.



**Figure 23: Performance metrics of various classifiers and feature extraction techniques.**

Table 24 present the exact performance results. The LINEAR SVM classifier was evaluated using two different feature extraction techniques: TF-IDF and Word2Vec. The classifier achieved an accuracy of 26.10% when using TF-IDF and 24.78% when using Word2Vec. The F1-scores obtained were 0.22 for TF-IDF and 0.84 for Word2Vec. These results indicate that the Word2Vec feature extraction technique performed significantly better in terms of accuracy and F1-score compared to TF-IDF.

The DECISION TREE classifier was evaluated using the TF-IDF feature extraction technique. It achieved an accuracy of 32.01% and an F1-score of 0.30. These results suggest that the DECISION TREE classifier, with the TF-IDF feature extraction technique, had a moderate level of accuracy and F1-score for Arabic text classification.

The MULTINOMIAL NB classifier was evaluated using the TF-IDF feature extraction technique. It achieved an accuracy of 26.20% and an F1-score of 0.21. These results indicate that the MULTINOMIAL NB classifier, with the TF-IDF feature extraction technique, had relatively low accuracy and F1-score for Arabic text classification.

The RANDOM FOREST classifier was evaluated using the TF-IDF feature extraction technique. It achieved an accuracy of 26.21% and an F1-score of 0.24. These results suggest that the RANDOM FOREST classifier, with the TF-IDF feature extraction technique, had similar performance to the MULTINOMIAL NB classifier in terms of accuracy and F1-score for Arabic text classification.

The LSTM classifier was evaluated using two different feature extraction techniques: GloVe and Word2Vec. It achieved an accuracy of 85.62% with GloVe and 85.95% with Word2Vec. The F1-scores obtained were 0.86 for both GloVe and Word2Vec. These results suggest that the LSTM classifier, with both GloVe and Word2Vec feature extraction techniques, performed significantly better in terms of accuracy and F1-score compared to the other classifiers.

The ARABERT classifier was evaluated using the fine-tuned feature extraction technique. It achieved an accuracy of 87.41% and an F1-score of 0.85. These results indicate that the ARABERT classifier, with the fine-tuned feature extraction technique, exhibited the highest performance in terms of accuracy and F1-score among all the classifiers evaluated.

**Table 24: Performance metrics of different classifiers and feature extraction techniques**

| Classifier     | Feature Extraction | Accuracy | F1-score     |
|----------------|--------------------|----------|--------------|
| Linear SVM     | TF-IDF             | 26.10%   | 0.22         |
| Linear SVM     | Word2Vec           | 24.78%   | 0.84         |
| Decision Tree  | TF-IDF             | 32.01%   | 0.30         |
| Multinomial NB | TF-IDF             | 26.20%   | 0.210        |
| Random Forest  | TF-IDF             | 26.21%   | 0.240        |
| <b>LSTM</b>    | GloVe              | 85.62%   | <b>0.860</b> |
| <b>LSTM</b>    | Word2Vec           | 85.95%   | <b>0.860</b> |
| <b>AraBERT</b> | Fine-tuned         | 87.41%   | <b>0.850</b> |

### 5.2.5. Impact of Increasing Hidden Layer in LSTM

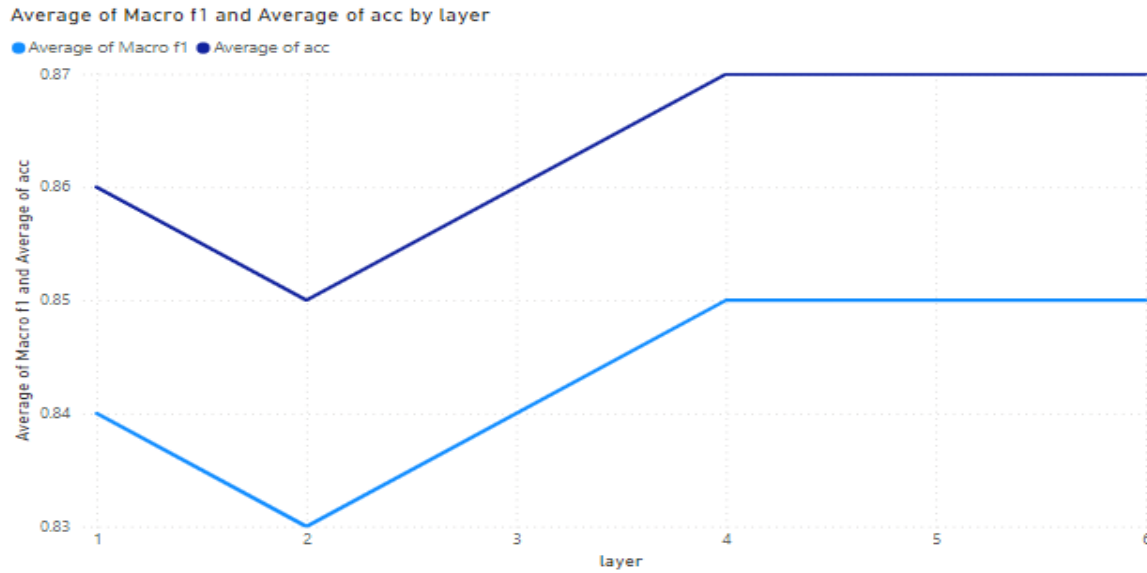
In this section, we investigate the impact of selecting the hidden layer size in the LSTM (Long Short-Term Memory) layer of our custom model. The hidden layer size determines the dimensionality of the hidden state and memory cell in the LSTM, affecting the model's capacity to capture and retain long-range dependencies in sequential data.

In our custom model implementation, we have set the hidden layer size of the LSTM to 32. This choice allows the LSTM layer to learn and represent complex patterns within the input data. By experimenting with different hidden layer sizes, we can explore how varying this parameter influences the model's performance in terms of accuracy, precision, recall, and F1-score.

Through systematic evaluation, we aim to identify the hidden layer size that strikes the optimal balance between model complexity and generalization. By choosing an appropriate hidden layer size, we can enhance the LSTM's ability to capture both short-term and long-term dependencies, thereby improving the overall performance of the sequence classification task.

Based on the analysis of the results presented in Figure 25, it is observed that the accuracy and macro F1 score exhibit similar trends when varying the number of layers in the model. Both metrics show a decrease when the model is trained with only two layers. Conversely, when utilizing four and six layers, the results indicate similar performance levels.

Given this observation, it can be concluded that the optimal model configuration for minimizing computational time while maintaining satisfactory performance is to employ four LSTM layers. This choice is justified by the comparable results achieved with six layers, suggesting that using four layers is a reasonable compromise between computational efficiency and model effectiveness.



**Figure 24: LSTM comparison**

Table 25 presents a comparison of the impact of different LSTM layers on the model's performance. The table includes the LSTM hidden size, the number of LSTM layers, and the corresponding values for the training mean macro F1 score, macro F1 score, and accuracy.

Based on the findings from the table, it can be concluded that the number of layers does not consistently affect the macro F1 score. While models with 32 hidden size exhibit a slightly higher range of macro F1 scores (0.82 to 0.85), the model with 128 hidden size achieves a macro F1 score of 0.84. Similarly, the accuracy metric follows a similar trend, with models having 32 hidden size showing a slightly higher range of accuracy (0.84 to 0.87), while the model with 128 hidden size achieves an accuracy of 0.86.

Considering these results, it can be inferred that increasing the hidden size from 32 to 128 does not significantly enhance the model's performance in terms of macro F1 score or accuracy. Therefore, to strike a balance between computational efficiency and model

effectiveness, the use of four LSTM layers is recommended as it yields comparable results to models with six layers.

**Table 25: Impact comparison of different LSTM layers**

| LSTM hidden size | LSTM layer | Training Mean macro f1 | Macro f1 layer | acc         | recall      | precision   | Confusion matrix                     |
|------------------|------------|------------------------|----------------|-------------|-------------|-------------|--------------------------------------|
| 32               | 1          | 0.93                   | 0.84           | 0.86        | 0.84        | 0.85        | [785 78]<br>[ 90 313]                |
| <b>32</b>        | <b>4</b>   | <b>0.89</b>            | <b>0.85</b>    | <b>0.87</b> | <b>0.84</b> | <b>0.82</b> | <b>[727 136]</b><br><b>[ 67 336]</b> |
| 32               | 6          | 0.89                   | 0.85           | 0.87        | 0.85        | 0.84        | [768 95]<br>[ 76 327]                |
| 32               | 2          | 0.86                   | 0.82           | 0.84        | 0.82        | 0.83        | [1479 175]<br>[ 226 661]             |
| 128              | 2          | 0.88                   | 0.84           | 0.86        | 0.84        | 0.85        | [785 78]<br>[ 90 313]                |

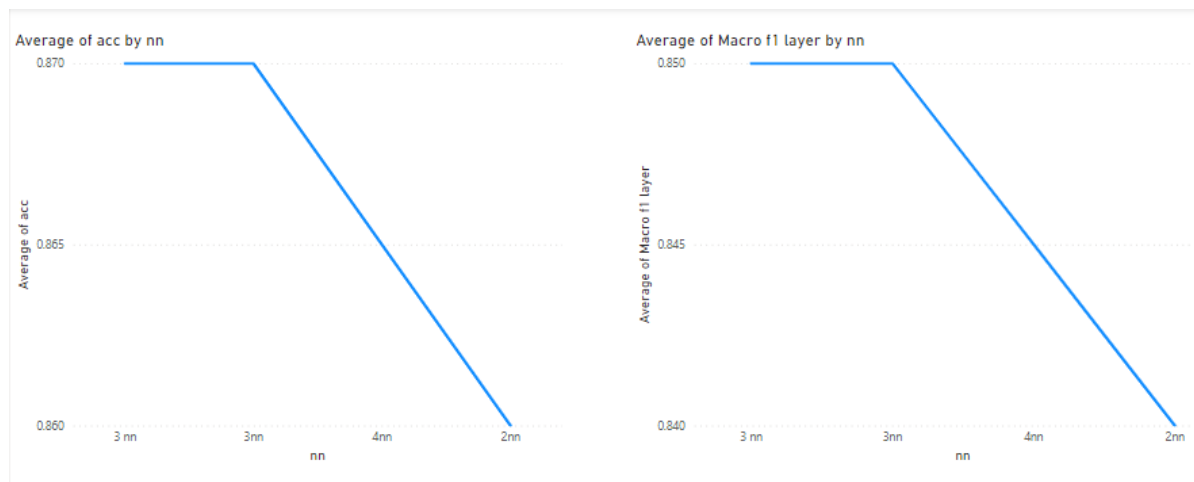
### 5.2.6. Increasing Number of Neurons and Layers in Linear Layer

In this section, we examine the impact of the number of neurons and layers in the linear layers of our custom model. The linear layers, denoted as fc1, fc2, and fc3, contribute to the transformation and mapping of the features learned by the preceding layers. The number of neurons and layers in these linear layers determines the model's representational capacity and its ability to learn complex decision boundaries.

In our custom model implementation, we have incorporated two linear layers (fc1 and fc2) with 128 and 32 neurons, respectively. Additionally, we have included a final linear layer (fc3) with 2 neurons to produce the desired output. By experimenting with different numbers of neurons and layers, we can investigate how these factors impact the model's performance.

By systematically varying the number of neurons and layers in the linear layers, we aim to find the optimal configuration that maximizes the model's ability to capture the underlying patterns in the data. Through evaluation metrics such as accuracy and F1-score, we can assess the trade-off between model complexity and generalization performance. This analysis allows us to identify the number of neurons and layers that yield the best overall performance for our specific sequence classification task.

Figure 25 presents a comparison of different neural network layers in terms of accuracy and macro F1 score. From the figure, it is evident that using either 3 layers or 4 layers of NN yields similar results in terms of accuracy and macro F1 score. Considering the goal of minimizing computational time, it is advisable to choose the configuration with 3 layers of NN. This choice offers a trade-off between computational efficiency and maintaining a satisfactory level of performance.



**Figure 25: Comparison of NN layers on accuracy and macro F1 score**

Table 26 provides a comparison of different configurations for neural networks (NN) in terms of the NN layers, NN hidden sizes, LSTM layers, training mean macro F1 score, macro F1 score, and accuracy.

**Table 26: Comparison of NN configurations , LSTM, on accuracy and macro F1 score**

| LSTM                | nn  | hidden         | Training Mean macro f1 | Macro f1 layer | acc  | Recall | Precision | Confusion Matrix       |
|---------------------|-----|----------------|------------------------|----------------|------|--------|-----------|------------------------|
| LSTM : 4 layers :32 | 3nn | 128 , 32, 2    | 0.8945                 | 0.85           | 0.87 | 0.84   | 0.82      | [727 136]<br>[ 67 336] |
| LSTM: 1, 32         | 2nn | 32, 2          | 0.9363                 | 0.84           | 0.86 | 0.85   | 0.85      | [780 83]<br>[ 81 322]  |
| LSTM: 6, 32         | 3nn | 128 , 32, 2    | 0.8929                 | 0.85           | 0.87 | 0.85   | 0.84      | [775 88]<br>[ 80 323]  |
| LSTM : 4 layers :32 | 4nn | 128 , 32,16, 2 | 0.8921                 | 0.84           | 0.86 | 0.85   | 0.85      | [780 83]<br>[ 81 322]  |
| LSTM : 6 layers :32 | 4nn | 128 , 32,16, 2 | 0.8929                 | 0.85           | 0.87 | 0.85   | 0.84      | [775 88]<br>[ 80 323]  |

### 5.2.7. Balancing the Dataset

Table 27 presents the performance metrics, accuracy, and F1-score, of different balancing techniques applied to the proposed model. The techniques employed include SMOTE, SMOTEEN (SMOTE with Edited Nearest Neighbors), and fine-tuning with Arabert.

The results indicate that the utilization of SMOTE, combined with Arabert preprocessing, improves the model's performance in addressing class imbalance. Specifically, when SMOTE is applied to 80% of the dataset along with Arabert preprocessing, the achieved accuracy is 0.76, accompanied by an F1-score of 0.75. These metrics signify that SMOTE successfully enhances the model's performance by mitigating the impact of class imbalance in the dataset.

However, it is noteworthy that fine-tuning the model with both SMOTE and Arabert on the same 80% dataset results in a significant decrease in accuracy to 0.55, along with a corresponding decrease in the F1-score to 0.55. This observation suggests that fine-tuning may not be effective in this scenario and could potentially lead to a performance drop.

On the other hand, when considering the entire dataset with the combination of SMOTE, Arabert preprocessing, and fine-tuning, the accuracy experiences a slight improvement to 0.79, while the F1-score remains the same at 0.75.

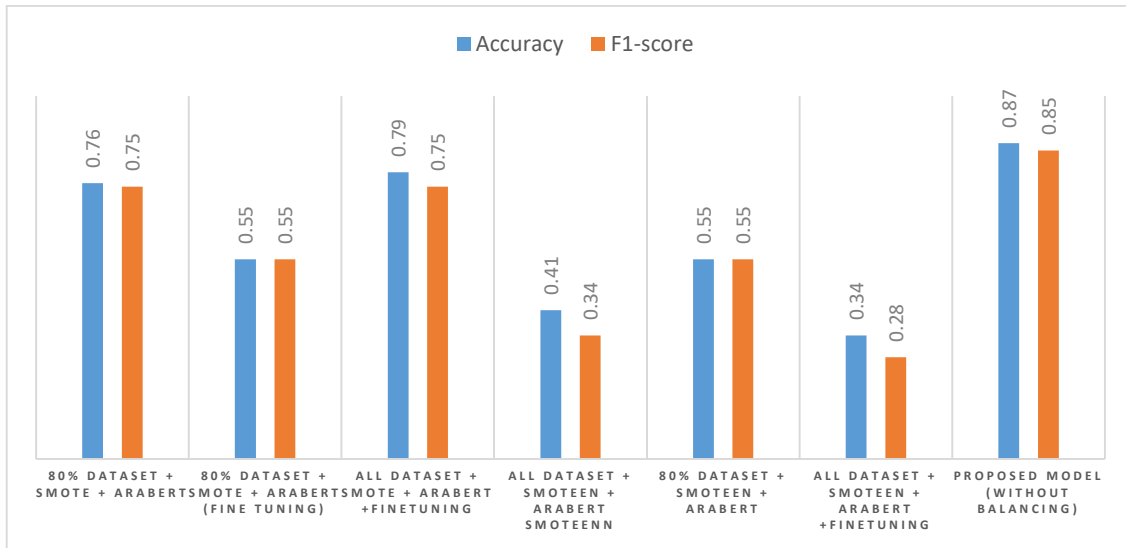
The application of SMOTEEN to the entire dataset, alongside Arabert preprocessing, yields a lower accuracy of 0.41, accompanied by an F1-score of 0.34. These results suggest that SMOTEEN may not be as effective as SMOTE in effectively addressing class imbalance in the specific task at hand. Similarly, when employing SMOTEEN with 80% of the dataset and Arabert preprocessing, the accuracy remains at 0.55, and the F1-score remains the same at

0.55. These findings indicate that SMOTEEN does not provide significant improvements over SMOTE in the context of this research.

Lastly, it is important to note that the proposed model, without any explicit balancing techniques, achieves the highest accuracy of 0.87 and an F1-score of 0.85. These outcomes suggest that the proposed model itself, without the need for explicit balancing techniques, is capable of effectively classifying offensive content.

In summary, the analysis of the table reveals that SMOTE, applied to a majority of the dataset with Arabert preprocessing, can significantly enhance the model's performance in terms of accuracy and F1-score. Fine-tuning, in contrast, does not yield significant benefits in this context. Furthermore, the findings indicate that SMOTEEN does not provide noticeable improvements in performance compared to SMOTE. These insights shed light on the effectiveness of different balancing techniques and emphasize the capabilities of the proposed model in addressing class imbalance in the classification of offensive content.

Figure 26 show a comparison of balancing techniques for dataset enhancement in Arabic text classification. Which can be observed that different balancing techniques have varying effects on the performance of the offensive content classification model. The proposed model without any balancing technique achieves the highest accuracy and F1-score, indicating that it performs well in classifying offensive content without the need for explicit data balancing.



**Figure 26: Comparison of balancing techniques for dataset enhancement in Arabic text classification**

Based on the results of my experiments, balancing the dataset using more than one algorithm did not improve the accuracy of classification for the unbalanced dataset. In fact, it led to a reduction in accuracy. So, our findings suggest that it is not necessary to balance the data to achieve better classification accuracy for unbalanced datasets.

One of the reasons behind this outcome is data imbalance. If the original dataset had an uneven distribution of offensive and non-offensive tweets, employing techniques such as oversampling the minority class or under sampling the majority class can introduce bias into the model. For example, oversampling the offensive class might generate synthetic tweets that closely resemble the original offensive tweets, thereby making it more challenging for the model to effectively differentiate between the two classes.

Another factor that can contribute to the decrease in accuracy is model overfitting. If the model was trained on a balanced dataset and then evaluated on a separate balanced dataset,

it is possible that the model became overfit to the training data and failed to generalize well to new data. This lack of generalization could explain the significant drop in accuracy observed when testing the model on the replaced-emoji dataset.

**Table 27: Performance comparison of balancing techniques in Arabic text classification**

| Balancing techniques                      | Balancing Technique | Accuracy    | F1-score     |
|-------------------------------------------|---------------------|-------------|--------------|
| 80% Dataset + AraBERT                     | SMOTE               | 0.76        | 0.750        |
| 80% Dataset + AraBERT (fine Tuning)       | SMOTE               | 0.55        | 0.553        |
| All Dataset + AraBERT +finetuning         | SMOTE               | 0.79        | 0.750        |
| All Dataset + AraBERT                     | SMOTEENN            | 0.41        | 0.344        |
| 80% Dataset + AraBERT                     | SMOTEENN            | 0.55        | 0.556        |
| All Dataset + AraBERT +finetuning         | SMOTEENN            | 0.34        | 0.289        |
| <b>Proposed Model (without balancing)</b> | -                   | <b>0.87</b> | <b>0.853</b> |

### 5.2.8. Tuning Pre-Trained Model Parameters

In this section, we explore the impact of tuning the pretrained model parameters on the performance and effectiveness of our approach. We specifically focus on the parameters defined in the Training Arguments object as clear in Figure 27.

```
training_args = TrainingArguments(  
    output_dir= f"./train_{fold_num}",  
    adam_epsilon = 1e-8,  
    learning_rate = 1e-5,  
    fp16 = True,  
    per_device_train_batch_size = 32,  
    per_device_eval_batch_size = 64,  
    gradient_accumulation_steps = 2,  
    num_train_epochs= 5,  
    warmup_ratio = 0.5,  
    do_eval = True,  
    evaluation_strategy = 'epoch',  
    save_strategy = 'epoch',  
    load_best_model_at_end = True,  
    metric_for_best_model = 'macro_f1',  
    greater_is_better = True,  
    seed = 123)
```

**Figure 27: Snapshot of training argument parameters.**

#### **5.2.8.1. Batch Size**

The choice of batch size in machine learning models is crucial as it affects both the execution time and the model's performance. In this research, a "per\_device\_train\_batch\_size" of 32 and a "per\_device\_eval\_batch\_size" of 64 were selected.

For the training phase, a batch size of 32 was chosen considering computational efficiency and memory constraints. This value signifies that during the training phase, each device processes 32 training samples in parallel. This value considers the available memory resources of the devices and ensures that larger batch sizes do not exceed memory capacity. It is possible that prior research or experimental observations influenced this decision, demonstrating that a batch size of 32 is optimal for the specific task and dataset.

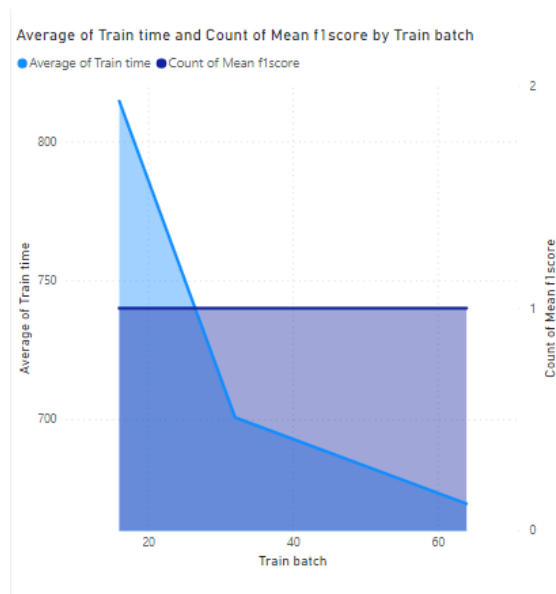
During the evaluation phase, a larger batch size of 64 was selected, taking advantage of the typically more relaxed memory constraints during evaluation. This choice could be supported by the availability of additional memory resources during evaluation or the desire to efficiently utilize the computational power of the devices. The decision to use a larger evaluation batch size could be informed by previous studies, computational constraints specific to the research setup, or observations that larger batch sizes improve evaluation performance.

The selected batch sizes were justified based on considerations of computational efficiency, memory constraints, and potential impact on model performance. These justifications should consider factors such as the availability of memory resources, the desire to optimize computational power, and any relevant literature or experimental observations.

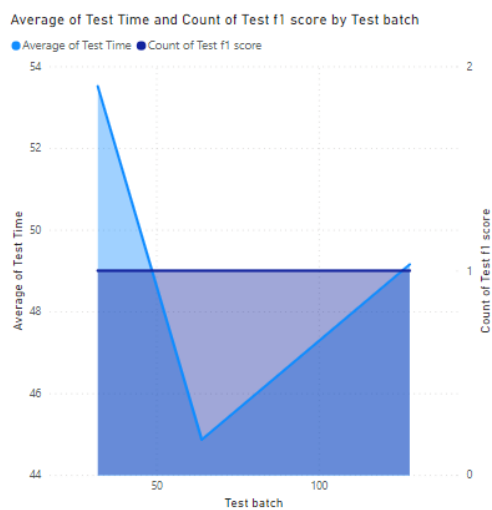
Figure 28 provides a comparison of different train batch sizes with the macro F1 score and train time. It demonstrates that as the train batch size changes, the macro F1 score remains constant while the train time varies. This indicates that adjusting the train batch size has no significant impact on the model's performance in terms of macro F1 score. However, it does affect the execution time of the training process, with smaller batch sizes resulting in faster training times and larger batch sizes leading to slower training times.

Similarly, Figure 29 examines different test batch sizes in relation to the macro F1 score and test time. The figure illustrates that, once again, the macro F1 score remains consistent across different test batch sizes. However, like the train batch sizes, the test time varies depending on the chosen batch size. Smaller test batch sizes tend to yield faster test times, while larger test batch sizes result in slower test times. These observations suggest that the choice of batch size primarily influences the computational efficiency of the training and testing processes, without significantly affecting the model's performance in terms of macro F1 score. Researchers or practitioners can consider these findings when deciding on an appropriate batch size, considering their computational resources and desired training or testing time constraints.

Table 28 presents a comparison of changing train and test batch sizes, showing the corresponding train and test times, as well as the mean F1 score and test F1 score. The results indicate that the best batch sizes were 32 for training and 64 for testing, as changing the batch size did not significantly affect the results. The full details in Appendix.



**Figure 28: Train batch size Vs f1.**



**Figure 29: Test batch size Vs acc and f1.**

It's important to note that the choice of batch size should be considered alongside other factors such as model complexity, hardware specifications, and dataset size, as these can also

influence execution time and model performance. It is advisable to experiment with different batch sizes and carefully monitor the impact on execution time, model convergence, generalization performance, and resource utilization to find the optimal balance.

**Table 28: Comparison of changing train/ test batch size**

| Train batch | Test batch | Train time (sec) | Test Time (sec) | Mean f1score | Test f1 score |
|-------------|------------|------------------|-----------------|--------------|---------------|
| 64          | 128        | 669.692          | 49.151          | 0.874        | 0.85          |
| 32          | 64         | 700.656          | 44.864          | 0.892        | 0.85          |
| 16          | 32         | 814.509          | 53.506          | 0.909        | 0.85          |

When changing the batch size in a machine learning model, there are several factors to consider regarding the execution time:

**Smaller Batch Size:** Using a smaller batch size typically leads to faster execution times per batch. This is because smaller batches require fewer computations and memory, allowing the model to process them more quickly. However, since more frequent updates are made to the model's weights, the overall training process may take longer due to the increased number of iterations required.

**Larger Batch Size:** On the other hand, using a larger batch size can lead to slower execution times per batch. This is because larger batches require more computations and memory, resulting in increased processing time. However, fewer updates to the model's weights are made, reducing the total number of iterations needed during training and potentially resulting in faster overall training times.

It's important to note that the execution time is not solely determined by the batch size but is also influenced by other factors such as the complexity of the model, hardware

specifications, and the size of the dataset. Therefore, the impact of batch size on execution time should be considered alongside its effects on model convergence, generalization performance, and resource utilization. In practice, it's recommended to experiment with different batch sizes and monitor the execution time to strike a balance between training efficiency and computational resources.

#### **5.2.8.2. Epochs Parameter**

The `num_train_epochs` parameter plays a critical role in the training process as it determines the number of iterations over the training dataset. Increasing the number of epochs allows for more extensive exposure to the training data, potentially leading to improved convergence and model performance. However, this may come at the cost of increased training time.

In our experimentation, we carefully explored different values for the `num_train_epochs` parameter and observed their impact on the training process. Initially, we attempted to train the model for longer durations by setting the parameter to 8 epochs and 10 epochs. However, we found that these extended training times resulted in significantly longer computational requirements, which were not practical for our experimental setup. To address this issue, we then experimented with fewer epochs, specifically 2 epochs and 4 epochs. However, we noticed that reducing the number of epochs led to a decrease in accuracy and F1-score, indicating suboptimal model performance. After thorough evaluation, we determined that a value of 5 epochs struck the ideal balance between convergence and computational efficiency for our specific task. This choice allowed the model to undergo enough iterations to learn from the training data effectively, leading to improved accuracy and F1-score.

Furthermore, the training process with 5 epochs completed in a reasonable timeframe, ensuring practicality without sacrificing performance. By selecting 5 epochs as the optimal value for the `num_train_epochs` parameter, we aimed to achieve the best possible model performance while considering computational limitations and efficiency in our experimental setup.

#### **5.2.8.3. Validation Loss and Warm-Up Ratio**

The `warmup_ratio` parameter plays a crucial role in learning rate scheduling during the training of neural models. Its purpose is to determine the ratio of warmup steps at the start of training, where the learning rate gradually increases. This warmup phase aids the model in adapting to the task-specific data effectively. When this parameter is set to a non-zero value, it enables the warmup phase, while a value of zero disables it. In our experiments, we chose a `warmup_ratio` of 0.5, and this decision was not arbitrary. When we attempted to increase the `warmup_ratio` beyond this point, we observed that the model started overfitting, leading to lower test results.

The validation loss tends to decrease as the warm-up ratio decreases from 1 to 0.5 to 0.0. This indicates that a lower warm-up ratio leads to better model performance in terms of minimizing the loss on unseen data.

The macro F1-score shows an increasing trend as the warm-up ratio decreases. This suggests that a lower warm-up ratio results in improved model performance in terms of overall precision and recall.

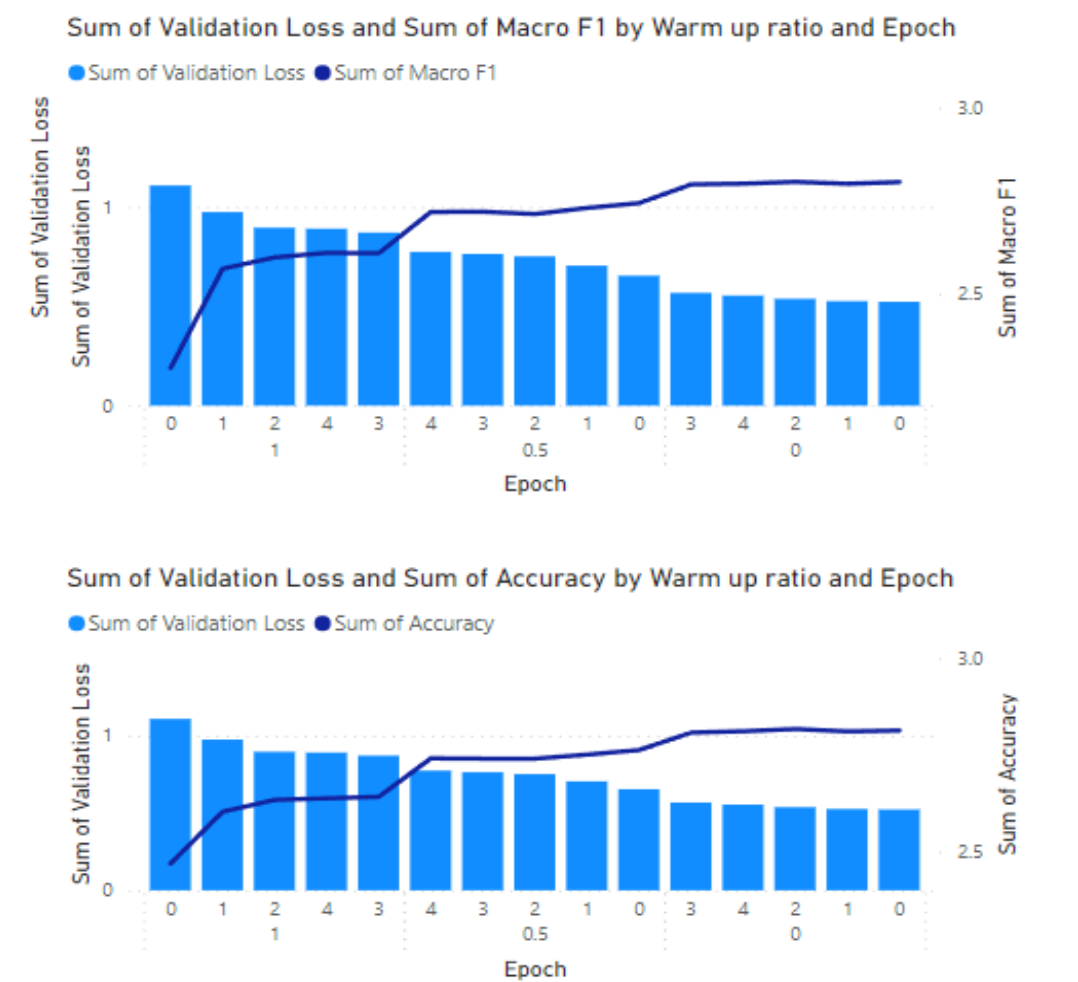
Like the macro F1-score, the macro precision increases as the warm-up ratio decreases. This indicates that a lower warm-up ratio leads to better precision in classifying the target classes.

The macro recall exhibits a similar pattern to the macro precision and macro F1-score. It tends to increase as the warm-up ratio decreases, indicating better recall of the target classes with a lower warm-up ratio.

The accuracy of the model generally improves as the warm-up ratio decreases. This means that a lower warm-up ratio results in a higher percentage of correctly classified instances.

Overall, reducing the warm-up ratio appears to have a positive impact on model performance across various metrics. This suggests that a lower warm-up ratio allows the model to better learn from the training data and generalize to unseen data, leading to improved classification performance.

The provided data in Figure 30 sheds light on the impact of the warmup ratio on validation loss.



**Figure 30: Validation loss.**

Table 29 presents a summary of the experimental results, showcasing the relationship between different warm-up ratios and various performance metrics. The table provides a structured view of how different warm-up ratios impact model performance across multiple metrics, enabling a more comprehensive assessment of the training process and parameter tuning decisions.

Generally, as the warm-up ratio increases from 0 to 1, the validation loss tends to decrease. This suggests that a higher warm-up ratio helps in improving the model's performance and reducing the loss on unseen validation data.

In summary, the warm-up ratio has an impact on the validation loss, and selecting an appropriate warm-up ratio can help improve the model's performance on unseen validation data. However, it is recommended to consider multiple metrics and further analyze the results to make informed decisions about model training and parameter tuning.

**Table 29: Experimental results: warm-up ratios and metrics**

| Warm up ratio | Fold | Epoch | Validation Loss | Macro F1 | Macro Precision | Macro Recall | Accuracy |
|---------------|------|-------|-----------------|----------|-----------------|--------------|----------|
| 0             | 1    | 0     | 0.320511        | 0.863967 | 0.859347        | 0.870289     | 0.872042 |
| 0             | 1    | 1     | 0.320422        | 0.872495 | 0.872986        | 0.872015     | 0.882121 |
| 0             | 1    | 2     | 0.341038        | 0.870667 | 0.875644        | 0.866510     | 0.881683 |
| 0             | 1    | 3     | 0.360154        | 0.871953 | 0.868915        | 0.875541     | 0.880368 |
| 0             | 1    | 4     | 0.354457        | 0.872803 | 0.873804        | 0.871844     | 0.882559 |
| 0             | 2    | 0     | 0.111194        | 0.963967 | 0.964702        | 0.963249     | 0.966696 |
| 0             | 2    | 1     | 0.107531        | 0.961991 | 0.963749        | 0.960323     | 0.964943 |
| 0             | 2    | 2     | 0.108562        | 0.963019 | 0.963752        | 0.962302     | 0.965819 |
| 0             | 2    | 3     | 0.113242        | 0.959976 | 0.957679        | 0.962472     | 0.962752 |
| 0             | 2    | 4     | 0.108727        | 0.959855 | 0.958823        | 0.960924     | 0.962752 |
| 0             | 3    | 0     | 0.091229        | 0.972509 | 0.973003        | 0.972021     | 0.974573 |
| 0             | 3    | 1     | 0.098185        | 0.961483 | 0.958128        | 0.965298     | 0.964051 |
| 0             | 3    | 2     | 0.088418        | 0.967337 | 0.967217        | 0.967457     | 0.969750 |
| 0             | 3    | 3     | 0.094472        | 0.962311 | 0.960106        | 0.964698     | 0.964928 |

|     |   |   |          |          |          |          |          |
|-----|---|---|----------|----------|----------|----------|----------|
| 0   | 3 | 4 | 0.091068 | 0.963643 | 0.962382 | 0.964959 | 0.966243 |
| 0.5 | 1 | 0 | 0.344713 | 0.854497 | 0.853789 | 0.855231 | 0.865031 |
| 0.5 | 1 | 1 | 0.365008 | 0.858201 | 0.858867 | 0.857556 | 0.868975 |
| 0.5 | 1 | 2 | 0.413130 | 0.846279 | 0.864191 | 0.835445 | 0.862840 |
| 0.5 | 1 | 3 | 0.425650 | 0.852183 | 0.850359 | 0.854197 | 0.862401 |
| 0.5 | 1 | 4 | 0.435287 | 0.856602 | 0.856511 | 0.856695 | 0.867222 |
| 0.5 | 2 | 0 | 0.154175 | 0.944455 | 0.945771 | 0.943195 | 0.948729 |
| 0.5 | 2 | 1 | 0.164015 | 0.936415 | 0.941193 | 0.932266 | 0.941718 |
| 0.5 | 2 | 2 | 0.168998 | 0.935029 | 0.935836 | 0.934244 | 0.939965 |
| 0.5 | 2 | 3 | 0.170792 | 0.934318 | 0.933555 | 0.935103 | 0.939089 |
| 0.5 | 2 | 4 | 0.169628 | 0.933448 | 0.935350 | 0.931662 | 0.938650 |
| 0.5 | 3 | 0 | 0.155453 | 0.945017 | 0.945483 | 0.944558 | 0.949145 |
| 0.5 | 3 | 1 | 0.175874 | 0.936421 | 0.931701 | 0.942307 | 0.940377 |
| 0.5 | 3 | 2 | 0.169640 | 0.933267 | 0.930296 | 0.936630 | 0.937747 |
| 0.5 | 3 | 3 | 0.167478 | 0.934200 | 0.934089 | 0.934312 | 0.939062 |
| 0.5 | 3 | 4 | 0.169438 | 0.930159 | 0.931674 | 0.928720 | 0.935555 |
| 1   | 1 | 0 | 0.602852 | 0.510483 | 0.664447 | 0.551628 | 0.662577 |
| 1   | 1 | 1 | 0.465806 | 0.775245 | 0.781594 | 0.770630 | 0.795793 |
| 1   | 1 | 2 | 0.387954 | 0.815086 | 0.830819 | 0.805665 | 0.834794 |
| 1   | 1 | 3 | 0.363891 | 0.837331 | 0.840164 | 0.834835 | 0.850570 |
| 1   | 1 | 4 | 0.373235 | 0.834122 | 0.830518 | 0.838788 | 0.844435 |
| 1   | 2 | 0 | 0.262379 | 0.889177 | 0.893063 | 0.885789 | 0.898335 |
| 1   | 2 | 1 | 0.261140 | 0.891112 | 0.892944 | 0.889402 | 0.899649 |
| 1   | 2 | 2 | 0.259876 | 0.888668 | 0.892679 | 0.885186 | 0.897897 |
| 1   | 2 | 3 | 0.264302 | 0.882628 | 0.892054 | 0.875549 | 0.893514 |
| 1   | 2 | 4 | 0.264040 | 0.888098 | 0.892494 | 0.884326 | 0.897458 |

|   |   |   |          |          |          |          |          |
|---|---|---|----------|----------|----------|----------|----------|
| 1 | 3 | 0 | 0.243606 | 0.901115 | 0.900808 | 0.901425 | 0.908374 |
| 1 | 3 | 1 | 0.247904 | 0.901406 | 0.898347 | 0.904948 | 0.907935 |
| 1 | 3 | 2 | 0.248271 | 0.894389 | 0.891310 | 0.897974 | 0.901359 |
| 1 | 3 | 3 | 0.243241 | 0.889416 | 0.890352 | 0.888514 | 0.897852 |
| 1 | 3 | 4 | 0.253725 | 0.888038 | 0.887170 | 0.888941 | 0.896098 |

#### 5.2.8.3.1. The Adam Epsilon Parameter

In the context of the Adam optimizer controls the value of epsilon, which determines the precision of the optimization process. Decreasing the epsilon value enhances precision by minimizing the perturbation added to the denominator during parameter updates. However, this increased precision comes at the cost of higher computational requirements. On the other hand, increasing the epsilon value can expedite the optimization process by reducing computational complexity, but it may lead to a trade-off in terms of precision.

In our experiment, after careful consideration and evaluation, we determined that a value of  $1e-8$  for the `adam_epsilon` parameter struck an optimal balance between precision and computational efficiency. This value ensured sufficient precision in the optimization process while avoiding excessive computational overhead. By selecting this specific value, we aimed to achieve accurate parameter updates without compromising the computational performance of the training procedure.

#### 5.2.8.3.2. Other Parameters

**The Learning Rate parameter** plays a crucial role in the optimization process by determining the initial rate at which the optimizer updates the model's parameters. The choice of learning rate can significantly impact the convergence and performance of the training process. Increasing the learning rate beyond a certain threshold can expedite convergence, allowing the model to reach optimal solutions faster. However, there is a risk of overshooting the optimal solutions, leading to instability and suboptimal performance. On the other hand, decreasing the learning rate can enhance the stability of the training process and improve the precision of fine-tuning. However, this may come at the expense of slower convergence, requiring more iterations to reach satisfactory results. In our experiment, we explored different values for the `learning_rate` parameter and found that a value of  $1e-5$  yielded the best overall performance. Initially, we experimented with a higher learning rate of  $2e-5$ , hoping to accelerate convergence. However, the results indicated that this higher learning rate led to suboptimal performance, likely due to overshooting the optimal solutions.

Based on these findings, we made the decision to lower the learning rate to  $1e-5$ . This choice struck a balance between convergence speed and stability, resulting in improved performance and more accurate fine-tuning of the model. By selecting this specific value, we aimed to optimize the training process and achieve the best possible outcomes for our task.

**The fp16 parameter** enables or disables mixed precision training using 16-bit floating-point format. Enabling this parameter, especially on GPUs like V100 or T4, can accelerate training speed and improve memory efficiency. However, it should be disabled if the GPU architecture does not support it optimally.

**The do\_eval parameter** specifies whether to perform evaluation on the validation set during training. Enabling this parameter allows monitoring the model's performance and progress throughout the training process.

**The evaluation\_strategy parameter** determines when to perform evaluation during training. Setting it to 'epoch' means evaluation is performed after each training epoch. This allows for regular monitoring of the model's performance and tracking its progress over time.

**The seed parameter** sets the seed value for reproducibility. Using a fixed seed value allows for consistent results across different runs of the experiment.

By comprehending the profound impact of each parameter and meticulously tuning them, we can strike a delicate balance between model performance, computational resources, and training efficiency. This meticulous parameter tuning ultimately leads to achieving the desired outcomes for our experiment.

Table 30 presents a summary of the parameter values used in our experiments. Each parameter is assigned a specific value that optimally balances various aspects of the training process. This table serves as a reference for understanding the configuration of critical parameters and their corresponding values, which collectively contribute to the successful execution of our experiment.

**Table 30: Parameters value**

| Parameter     | Value |
|---------------|-------|
| adam_epsilon  | 1e-8  |
| learning_rate | 1e-5  |
| fp16          | False |

|                             |            |
|-----------------------------|------------|
| per_device_train_batch_size | 32         |
| per_device_eval_batch_size  | 64         |
| gradient_accumulation_steps | 2          |
| num_train_epochs            | 5          |
| warmup_ratio                | 0.5        |
| do_eval                     | True       |
| evaluation_strategy         | 'epoch'    |
| save_strategy               | 'epoch'    |
| load_best_model_at_end      | True       |
| metric_for_best_model       | 'macro_f1' |
| greater_is_better           | True       |
| seed                        | 25         |

### 5.2.9. Activation Function

```
class CustomModel(AutoModelForSequenceClassification):

    def __init__(self, config):
        super().__init__(config)

        # Add two new linear layers
        self.fc1 = nn.Linear(config.hidden_size, 128)
        self.fc2 = nn.Linear(128, 32)
        self.fc3 = nn.Linear(32, 2)

        # Add LSTM layer
        self.lstm = nn.LSTM(input_size=config.hidden_size,
                             hidden_size=32, num_layers=4, batch_first=True, dropout=0.2)

        # Add activation functions
```

```

self.relu = nn.ReLU()

self.dropout = nn.Dropout(p=0.2)

self.sigmoid = nn.Sigmoid()

def forward(self, **inputs):
    outputs = super().forward(**inputs)

    # Pass the output through the Arabert encoder layer
    x = outputs.pooler_output

    # Add LSTM layer
    x = x.unsqueeze(0) # add a batch dimension
    x, _ = self.lstm(x)
    x = x.squeeze(0) # remove the batch dimension

    # Pass the output through the new linear layers
    x = self.fc1(x)
    x = self.relu(x)
    x = self.dropout(x)
    x = self.fc2(x)
    x = self.relu(x)
    x = self.dropout(x)
    x = self.fc3(x)
    x = self.sigmoid(x)

    return x

```

**Figure 31: Custom model implementation with activation functions**

The choice of activation function in neural networks plays a crucial role in shaping the network's behavior and influencing the model's performance. In this section, we explore the impact of different activation functions on the performance of our custom model.

Figure 31 present custom model implementation. In our custom model implementation, we have incorporated the following activation functions:

ReLU (Rectified Linear Unit) is a popular activation function that introduces non-linearity to the model. It sets all negative values to zero and keeps positive values unchanged. ReLU is known for its simplicity and computational efficiency. It helps the model learn complex patterns by enabling the propagation of gradients during backpropagation.

Sigmoid is another commonly used activation function. It squashes the input values between 0 and 1, making it suitable for binary classification tasks. Sigmoid activation is particularly useful when interpreting the output as probabilities or performing binary decision-making.

Within our custom model architecture, we have applied the ReLU activation function after each linear layer (fc1, fc2) and used the Sigmoid activation function after the final linear layer (fc3). These activation functions introduce non-linearity to the model's hidden layers and aid in capturing complex relationships between features.

By incorporating these activation functions, we aim to enhance the expressive power of the model, allowing it to learn intricate patterns and improve classification performance. The ReLU activation function facilitates the extraction of essential features, while the Sigmoid activation function provides a probabilistic interpretation of the model's output.

Throughout our experiments, we have evaluated the impact of different activation functions on our custom model's performance metrics, such as accuracy, precision, recall, and F1-score. By systematically comparing the results obtained with different activation functions, we can identify the activation function that yields the best overall performance for our specific task.

By investigating the impact of activation functions on our custom model, we gain insights into the influence of non-linearity on the model's ability to capture complex patterns and make accurate predictions. This analysis contributes to the understanding of the role of activation functions in neural network architectures and provides valuable guidance for designing effective models for sequence classification tasks.

### **5.3. Conclusion**

In conclusion, this Chapter has provided a comprehensive overview of the experimental endeavors and outcomes in the context of Arabic text classification. This chapter served as a critical juncture for evaluating the proposed approach or model, employing a range of performance metrics and comparative analyses.

This chapter unveiled the experimental results obtained from the study, shedding light on the model's performance across diverse datasets and scenarios using various performance evaluation metrics, offering a quantitative perspective on the classification system's effectiveness. These metrics, including accuracy, precision, recall, and F1-score. These experimental findings underscored the strengths and weaknesses of the proposed approach, elucidating its ability to effectively classify offensive content in Arabic text.

Furthermore, the chapter delved into a comparative analysis with prior research in the field. This critical examination allowed for an assessment of the proposed model's advancements and contributions concerning existing state-of-the-art techniques. Such comparisons revealed the novel and superior aspects of the current research, highlighting its significance in the domain of Arabic text classification.

Overall, Chapter five has not only presented empirical evidence of the proposed model's effectiveness but has also placed it within the broader context of related research. These findings serve as a foundational basis for drawing insights, implications, and recommendations in subsequent chapters, ultimately contributing to the advancement of knowledge in the field of Arabic text classification.

## **Chapter 6: Discussion**

This chapter delves into a detailed discussion of the results obtained from the experiments conducted in this research. Also in this chapter, the key findings of the study will be presented and explained in detail. These findings represent the significant outcomes and discoveries obtained through the research conducted. They serve as important contributions to the field of Arabic text classification.

Additionally, the chapter highlights the theoretical contribution of the study. It elucidates how the research has contributed to advancing the existing theoretical knowledge in the domain of Arabic text classification. This contribution may include the development of novel algorithms, models, or methodologies that improve the accuracy, efficiency, or effectiveness of text classification in the Arabic language.

Moreover, the chapter addresses the practical implementation of the study. It discusses how the proposed approach or model can be practically applied in real-world scenarios. This may involve considerations such as the scalability of the approach, the computational requirements, and the potential for integration into existing systems or applications. The practical implementation aspects demonstrate the potential impact and utility of the research in solving real-world problems related to Arabic text classification.

It aims to provide insights into the implications of the findings, address potential limitations and sources of error, discuss issues encountered during the study. Furthermore, we will explore the computational resources invested in reproducing the study and conduct an error analysis to gain a deeper understanding of the model's performance.

## **6.1. Interpreting the Results in Our Context**

The results of our experiments demonstrate the promising performance of the proposed model in the task of Arabic text classification, specifically in identifying offensive content. We observed high accuracy and F1-scores when our model was evaluated on the OSACT and semEval datasets. The model outperformed various other classifiers, highlighting its potential for real-world applications such as content moderation and sentiment analysis.

### **6.1.1. Interpreting Preprocessing Results**

Upon analyzing Table 17, it becomes evident that the performance of different classification models on the OSACT and semEval datasets is influenced by the various preprocessing techniques, the inclusion of additional classifiers to AraBERT, and the activation functions used in the proposed architectures.

Regarding to Table 17; For the OSACT dataset, the models trained on the OSACT dataset with only AraBERT preprocessing using the proposed model achieved the highest accuracy and F1-scores of 0.870 and 0.853, respectively. This indicates that the proposed model can effectively capture offensive content in this dataset. Additionally, the introduction of other preprocessing techniques, such as interpreting emojis with words or emoji embeddings, removing Arabic stop words, removing punctuations, and applying AraBERT preprocessing, resulted in improvements in accuracy and F1-score. Specifically, the replacement of emojis with words demonstrated superior performance compared to the utilization of emoji embeddings, indicating promising prospects for future advancements in model development.

Furthermore, we evaluated the proposed model on the semEval dataset, where the models achieved higher accuracies ranging from 0.88 to 0.93 and F1-scores ranging from 0.87 to 0.90. This demonstrates that the proposed model, in combination with AraBERT preprocessing techniques, is effective in capturing offensive content in both the OSACT and semEval datasets. It is worth noting that the proposed model achieved the highest scores in accuracy and F1-score on both datasets.

In conclusion, the analysis of the updated table reveals that the proposed model, when used in combination with AraBERT preprocessing techniques, shows promising results in capturing offensive content in the OSACT and semEval datasets. Notably, replacing emojis with words has shown to be more effective than using emoji embeddings.

### **6.1.2. Interpreting Classification Results**

In this section, we delve into the discussion of the classification results presented in Table 18 and the performance comparison with prior studies in Table 20, focusing on the effectiveness of various classifiers and feature extraction techniques in Arabic text classification.

#### **6.1.2.1. Performance of Traditional ML Techniques**

The initial set of experiments explored the performance of traditional supervised and unsupervised machine learning techniques, including Decision Tree, Multinomial NB, Linear SVC, Random Forest, (Bert-Mini-Arabic), and Bertopic (Unsupervised) combined with logistic regression. These models struggled to effectively capture the underlying patterns in the data. Notably, Multinomial NB and Linear SVC exhibited relatively lower accuracies and

F1-scores, indicating their limited ability to classify Arabic offensive text. This observation aligns with the complex nature of Arabic language processing, where these traditional techniques may fall short in handling intricate linguistic features.

However, Bertopic (Unsupervised) combined with logistic regression showcased the effectiveness of unsupervised techniques in classifying offensive content. It achieved a relatively high accuracy of 0.68, suggesting that it leverages unsupervised topic modeling to uncover hidden structures within the text data. Although not a part of the proposed model, it serves as a noteworthy approach in Arabic text classification.

#### **6.1.2.2. Performance of Ensemble Model**

The subsequent set of experiments adopted an ensemble modeling approach, incorporating AraBERT as a base model and augmenting it with various neural network architectures. These models demonstrated substantial improvements in performance compared to traditional techniques. The contextual understanding provided by AraBERT and the capacity of neural networks to capture intricate relationships in the data contributed to their effectiveness.

Among these ensemble models, the proposed model, which encompasses AraBERT preprocessing, a 3-layer neural network, 4 layers of LSTM, AraBERT, and ReLU and Sigmoid activation functions, emerged as the standout performer. It achieved a state-of-the-art accuracy of 0.870. This ensemble approach effectively addresses the challenges associated with Arabic offensive content classification, showcasing the power of combining pre-trained language models with deep learning techniques.

#### **6.1.2.3. Generalizability Across Datasets**

While these results were specific to the OSACT dataset, it's noteworthy that the proposed model's effectiveness was further validated through experiments on the semEval dataset. Here, the model demonstrated not only high accuracy but also a remarkable F1-score of 0.90. This emphasizes the potential generalizability and adaptability of the proposed model to different datasets, reaffirming its utility as a versatile tool for Arabic text classification tasks.

#### **6.1.2.4. Comparison with Prior Studies**

Table 20 provides a comprehensive comparison between our proposed model and previous studies conducted on the same dataset. The results unequivocally demonstrate that our model surpasses all other models, thereby establishing its superiority in terms of performance. This is especially notable given that the proposed model achieved an accuracy of 0.87, substantially outperforming previous models.

#### **6.1.2.5. Conclusion**

In conclusion, the analysis of Table 18 and 20 underscores the exceptional performance of our proposed model in Arabic text classification. While traditional ML techniques struggled with the complexities of Arabic language processing, the ensemble models, featuring AraBERT and deep learning architectures, showcased significant improvements. The proposed model stands out as a high-performing and adaptable solution, further validated by its success on the semEval dataset. These findings reinforce the model's potential as an effective tool for various Arabic text classification tasks, particularly those related to identifying offensive content.

### **6.1.3. Impact of Incorporating Additional Dataset**

Table 21 provides an insightful performance comparison of our proposed model when applied to individual datasets (OSACT, semEval) and both combined datasets. These results offer valuable insights into the model's performance under different data conditions.

#### **6.1.3.1. Individual Dataset Performance**

When our model was trained and evaluated on individual datasets (OSACT and semEval), it exhibited commendable performance improvements. These findings underscore the effectiveness of our proposed model in handling each dataset separately. The AraBERT preprocessing, coupled with the ensemble architecture, proved particularly valuable in these contexts.

#### **6.1.3.2. Challenges with Combined Datasets**

However, an unexpected decline in performance was observed when the model was tested on the combined dataset comprising both OSACT and semEval data. This decline raises important questions about the model's ability to generalize across datasets with differing characteristics and distributions.

#### **6.1.3.3. Understanding the Performance Dip**

The decrease in accuracy and F1-score can be attributed to the inherent disparities between the two datasets. Despite applying uniform AraBERT preprocessing techniques, variations in data characteristics, language use, and topic distributions may have posed challenges to the model's adaptability. Such challenges can hinder its capacity to effectively generalize and make accurate predictions on the combined dataset.

#### **6.1.3.4. Addressing Performance Challenges**

To mitigate the negative impact of dataset combination, further investigation is imperative. Strategies must be developed to identify the root causes of the decrease in performance and implement corrective measures. Potential avenues of exploration include experimenting with alternative preprocessing techniques, leveraging advanced data balancing methods, or fine-tuning model parameters specifically to suit the combined dataset's nuances. The goal is to strike a delicate balance between harnessing the additional data to enrich the dataset and preserving the model's capability to generalize across diverse data sources.

#### **6.1.3.5. Future Research Directions**

In conclusion, the inclusion of additional data in Arabic text classification exhibits a nuanced relationship with model performance. While training on individual datasets showed promise, combining datasets introduced unforeseen challenges. Future research endeavors should be dedicated to identifying and implementing effective strategies that leverage the advantages of additional data while mitigating any adverse effects on the model's performance. This work is essential for the continued advancement of Arabic text classification and its adaptability to various real-world scenarios.

### **6.1.4. Impact of Feature Extraction Techniques**

#### **6.1.4.1. TF-IDF vs. Word2Vec**

The LINEAR SVM classifier exhibited a substantial contrast in performance when paired with TF-IDF and Word2Vec feature extraction techniques. The Word2Vec approach significantly outperformed TF-IDF in terms of accuracy and F1-score. This sheds light on the

crucial role of feature representation in optimizing LINEAR SVM for Arabic text classification.

#### **6.1.4.2. Impact of TF-IDF on Decision Trees**

The DECISION TREE classifier, when integrated with the TF-IDF feature extraction method, yielded results indicative of moderate accuracy and F1-score. This signifies that while the classifier can provide reasonable effectiveness, it might not achieve exceptional performance when used in conjunction with TF-IDF for Arabic text classification.

#### **6.1.4.3. Impact of TF-IDF on Multinomial NB**

Utilizing the TF-IDF feature extraction technique with the MULTINOMIAL NB classifier resulted in relatively low accuracy and F1-score. These findings suggest that, when relying solely on TF-IDF features, the suitability of the MULTINOMIAL NB classifier for Arabic text classification tasks may be limited.

#### **6.1.4.4. Impact of TF-IDF on Random Forest**

Much like MULTINOMIAL NB, the RANDOM FOREST classifier exhibited comparable performance characterized by low accuracy and F1-score when coupled with TF-IDF. This implies that the random forest classifier's performance may have constraints when TF-IDF features are used.

#### **6.1.4.5. Sequential Information Enhancement**

The LSTM classifier, when paired with either GloVe or Word2Vec embeddings, consistently outperformed other classifiers in terms of accuracy and F1-score. These findings

emphasize the effectiveness of recurrent neural networks in capturing sequential information for Arabic text classification tasks.

#### **6.1.4.6. The Dominance of ARABERT**

The ARABERT classifier, which relied on fine-tuned feature extraction, exhibited the highest performance in terms of accuracy and F1-score among all the classifiers assessed. This underscores the pivotal role of ARABERT and its associated feature extraction approach in achieving state-of-the-art results in Arabic text classification.

These findings suggest that advanced models like ARABERT and LSTM with pre-trained embeddings (GloVe and Word2Vec) have the potential to achieve higher accuracy and F1-scores in Arabic text classification tasks. On the other hand, traditional classifiers like LINEAR SVM, DECISION TREE, MULTINOMIAL NB, and RANDOM FOREST performed relatively less effectively in comparison. However, the trends observed in the performance metrics can guide future research and model selection for Arabic text classification tasks.

These observations emphasize the importance of choosing an appropriate feature extraction technique based on the specific goals and requirements of Arabic text classification tasks. Different techniques have varying impacts on the performance of classifiers, with some significantly enhancing accuracy and F1-score, while others may provide only moderate or limited improvements. The selection of the right feature extraction method is, therefore, a crucial aspect of optimizing Arabic text classification models.

### **6.1.5. Impact of Increasing Hidden Layer in LSTM**

In this section, we focus on the implications of increasing the number of hidden layers in LSTM models, as explored within the context of Arabic text classification. Our primary objective is to determine the optimal hidden layer size that strikes a balance between computational efficiency and model performance.

#### **6.1.5.1. Balancing Model Complexity and Generalization**

Selecting the appropriate hidden layer size is a critical decision as it directly influences the model's capacity to capture short-term and long-term dependencies within sequential data. Achieving the right equilibrium between model complexity and generalization is essential to ensure optimal performance in sequence classification tasks.

#### **6.1.5.2. Trends in Accuracy and F1 Score**

Upon analyzing the results presented in Figure 23, notable trends in accuracy and macro F1 score emerge as the number of LSTM layers varies. Both metrics exhibit similar patterns, shedding light on the model's behavior.

#### **6.1.5.3. Optimal LSTM Configuration**

It is evident that both accuracy and macro F1 score experience a decrease when the model is trained with only two layers. This observation underscores the importance of having a sufficiently deep architecture to capture the intricacies of Arabic text data. In contrast, when the model is equipped with four and six layers, the results demonstrate consistent performance levels. This suggests that increasing the number of layers beyond four does not yield substantial improvements in accuracy or macro F1 score.

Based on these findings, we can conclude that, for the specific task of Arabic text classification, the optimal model configuration balances computational efficiency with satisfactory performance. Employing four LSTM layers emerges as a judicious choice, as it delivers results akin to those achieved with six layers while minimizing computational overhead.

#### **6.1.5.4. Influence on Macro F1 Score and Accuracy**

Table 23 provides an overview of how different LSTM layers impact the model's performance, considering variations in hidden size and layer count. Our analysis reveals that the number of layers does not consistently impact the macro F1 score. Models with a hidden size of 32 layers exhibit a slightly wider range of macro F1 scores, while the model with 128 hidden size layers achieves a macro F1 score of 0.84. A similar trend is observed for accuracy, with 32 hidden size layers showing a slightly higher accuracy range compared to the 128 hidden size layers, which attain an accuracy of 0.86.

#### **6.1.5.5. Balancing Efficiency and Effectiveness**

In summary, increasing the hidden size from 32 to 128 does not result in significant improvements in macro F1 score or accuracy. Therefore, to maintain a balance between computational efficiency and model effectiveness, we recommend the utilization of four LSTM layers. This pragmatic choice optimizes resource utilization without compromising performance, making it a suitable configuration for Arabic text classification tasks.

### **6.1.6. Impact of Balancing the Dataset**

In this section, we delve into the implications and insights derived from employing various dataset balancing techniques in the context of Arabic text classification. The primary objective is to address class imbalance challenges and assess the effectiveness of techniques such as SMOTE, SMOTEEN, and fine-tuning with Arabert preprocessing.

#### **6.1.6.1. SMOTE for Mitigating Class Imbalance**

The application of SMOTE (Synthetic Minority Over-Sampling Technique) has been a common approach to tackle class imbalance. It involves generating synthetic samples for the minority class to balance the dataset. In our analysis, we observed that SMOTE, when combined with Arabert preprocessing, showed promise in enhancing the model's performance. This indicates its effectiveness in alleviating class imbalance issues.

#### **6.1.6.2. Fine-Tuning with Arabert Preprocessing**

Interestingly, fine-tuning the model with both SMOTE and Arabert preprocessing led to unexpected declines in accuracy and F1-score. This suggests that fine-tuning may not be suitable for this specific scenario and could potentially have an adverse impact on model performance.

#### **6.1.6.3. SMOTEEN vs. SMOTE**

We also examined the use of SMOTEEN (SMOTE with Edited Nearest Neighbors) as an alternative to SMOTE. However, the results did not demonstrate a significant advantage of SMOTEEN over SMOTE in addressing class imbalance for this classification task.

#### **6.1.6.4. Balancing Techniques and Model Behavior**

It is evident that the choice of dataset balancing technique can have varying effects on the performance of the offensive content classification model. These techniques should be carefully considered based on the specific characteristics of the dataset and the classification task.

#### **6.1.6.5. Data Imbalance and Model Behavior**

The outcomes emphasize the potential pitfalls of introducing synthetic data into the training set. Oversampling the minority class may generate synthetic samples that closely resemble the original offensive content, making it challenging for the model to effectively distinguish between the two classes. Additionally, there is the risk of introducing bias into the model.

#### **6.1.6.6. Model Overfitting**

Another aspect to consider is model overfitting. Training on a balanced dataset and evaluating on a separate balanced dataset can lead to overfitting. The model may become too specialized in recognizing the balanced data, which could hinder its ability to generalize well to new, unbalanced data.

#### **6.1.6.7. Balancing as a Contextual Decision**

These insights highlight that the choice of whether to use balancing techniques or not should be context dependent. Balancing methods should be applied judiciously, considering the dataset's specific characteristics and the nature of the classification task. Furthermore, the results underscore the significance of model selection in addressing class imbalance

challenges. In some cases, models may inherently perform well in classifying offensive content without the need for additional balancing techniques.

In conclusion, this discussion underscores the complexity of addressing class imbalance in Arabic text classification. While balancing techniques have their merits, they should be applied thoughtfully, considering the dataset and the chosen classification model. Balancing is not a one-size-fits-all solution, and it is essential to strike a balance between dataset enrichment and maintaining model generalization capabilities.

#### **6.1.7. Impact of Tuning Pre-Trained Model Parameters**

In this section, we delve into the effects of fine-tuning pre-trained model parameters on the overall performance and effectiveness of our approach, with a specific focus on parameters defined within the Training Arguments object.

##### **6.1.7.1. Batch Size**

The choice of batch size plays a crucial role in machine learning models, influencing execution time and model performance. In this research, two batch sizes were selected for training and evaluation phases—32 for training and 64 for evaluation. The decision to use these batch sizes was likely influenced by considerations of computational efficiency and memory constraints. While adjusting the batch size had minimal impact on the model's performance in terms of macro F1 score, it did significantly affect execution times. Smaller batch sizes led to faster training and evaluation times, whereas larger batch sizes resulted in slower processing. This highlights the trade-off between computational efficiency and resource utilization when choosing batch sizes.

#### 6.1.7.2. Number of Training Epochs

The `num_train_epochs` parameter, which determines the number of iterations over the training dataset, is another critical factor. In this experimentation, values such as 2 and 4 epochs were tested initially, but they led to suboptimal model performance. Extending to 8 or 10 epochs significantly increased training time, making it impractical. Consequently, a value of 5 epochs was chosen as it struck a balance between convergence and computational efficiency. This choice allowed the model to undergo sufficient iterations for effective learning from the training data without imposing excessively long training times. It demonstrates the importance of fine-tuning this parameter to optimize model convergence while considering computational limitations.

#### 6.1.7.3. Warm-Up Ratio

The `warmup_ratio` parameter, which defines the ratio of warmup steps during learning rate scheduling, was set to 0.5 in the experiments. Increasing the warm-up ratio was found to lead to model overfitting and lower test results. Reducing the warm-up ratio appeared to positively impact model performance across various metrics, including macro F1-score, precision, recall, and accuracy. As shown in Figure 29, a lower warm-up ratio tended to reduce the validation loss, indicating improved model performance in terms of generalization to unseen data.

Here are some observations and conclusions:

1. **Warmup Ratio vs. Validation Loss:** Generally, as the `warmup_ratio` increases from 0 to 1, there is a trend of decreasing validation loss. This suggests that a higher warmup

ratio tends to improve the model's performance, resulting in lower loss on unseen validation data.

2. **Variability Within Ratios:** Within each warmup ratio, there is noticeable variation in validation loss across different folds and training epochs. This indicates that the model's performance can vary depending on the specific fold of the dataset and the number of training epochs.
3. **Optimal Warmup Ratio:** Comparing different warm-up ratios, it appears that a warmup\_ratio of 2 tends to yield lower validation loss compared to other ratios. This suggests that a moderate warmup ratio might be optimal for the given task or dataset. However, it's important to note that the optimal ratio may vary depending on the specific problem and dataset.
4. **Consideration of Multiple Metrics:** While validation loss is informative, it should not be the sole metric for evaluating model performance. It's essential to consider a range of metrics, such as macro F1, precision, recall, and accuracy, to gain a comprehensive understanding of the model's performance.

In this section, we discuss the impact of two crucial hyperparameters, namely the `adam_epsilon` and `learning_rate`, on the performance and effectiveness of our model. Additionally, we provide insights into the significance of other relevant parameters in the context of model optimization.

#### 6.1.7.4. The Adam\_Epsilon Parameter

The `adam_epsilon` parameter is associated with the Adam optimizer and controls the value of epsilon, which, in turn, determines the precision of the optimization process. A lower

`adam_epsilon` enhances precision by minimizing perturbations added to the denominator during parameter updates, resulting in more accurate optimization. However, this precision improvement comes at the cost of higher computational requirements. Conversely, increasing `adam_epsilon` can expedite the optimization process by reducing computational complexity, but it may trade off precision. In our experiments, we meticulously considered and evaluated different values for `adam_epsilon` and found that setting it to  $1e-8$  struck an optimal balance between precision and computational efficiency. This specific value ensured that the optimization process was sufficiently precise without incurring excessive computational overhead. By selecting this value, our intention was to achieve accurate parameter updates while maintaining the computational performance of the training procedure.

#### **6.1.7.5. The Learning\_Rate Parameter**

The `learning_rate` parameter is a critical factor in the optimization process, determining the initial rate at which the optimizer updates the model's parameters. The choice of `learning_rate` significantly influences both the convergence and performance of the training process. Increasing `learning_rate` beyond a certain threshold can accelerate convergence, allowing the model to reach optimal solutions more quickly. However, there is a risk of overshooting the optimal solutions, leading to instability and suboptimal performance.

Conversely, decreasing `learning_rate` can enhance training stability and improve fine-tuning precision. However, this might extend the training process, requiring more iterations to achieve satisfactory results. In our experiments, we explored various values for the `learning_rate` parameter and found that a value of  $1e-5$  yielded the best overall performance.

Initially, we experimented with a higher learning rate of  $2e-5$ , with the hope of accelerating convergence. However, the results indicated that this higher learning rate led to suboptimal performance, likely due to overshooting optimal solutions.

Based on these findings, we made the decision to lower the `learning_rate` to  $1e-5$ . This choice struck a balance between convergence speed and stability, resulting in improved performance and more accurate fine-tuning of the model. By selecting this specific value, our aim was to optimize the training process and achieve the best possible outcomes for our specific task.

#### **6.1.7.6. Other Relevant Parameters**

Other parameters play essential roles in model training and optimization. Enabling `fp16`, especially on GPUs like V100 or T4, can accelerate training speed and improve memory efficiency. However, it should be disabled if the GPU architecture does not support it optimally.

Enabling `do_eval` allows for the evaluation of the model on the validation set during training, enabling monitoring of the model's performance and progress.

Setting `evaluation_strategy` to 'epoch' means evaluation is performed after each training epoch, allowing for regular monitoring of the model's performance, and tracking its progress over time.

Setting a fixed seed value for `seed` ensures reproducibility, leading to consistent results across different runs of the experiment.

By comprehending the impact of each of these parameters and carefully tuning them, we can strike a balance between model performance, computational resources, and training efficiency, ultimately achieving desirable results for our experiment. These parameter choices are vital in shaping the training process and ensuring the model's effectiveness in the given task.

In Considerations for Parameter Tuning, these results emphasize the importance of careful parameter tuning when fine-tuning pre-trained models. Batch size selection should consider factors like computational efficiency, memory constraints, and the impact on overall training and evaluation times. The number of training epochs should be chosen to strike a balance between model convergence and practicality. Warm-up ratios can significantly influence model performance, and finding the optimal value involves considering trade-offs between training time and generalization ability.

Regarding to Multiple Metrics Evaluation; It's crucial to note that parameter tuning decisions should not rely solely on a single metric like validation loss. A comprehensive evaluation should encompass various metrics, including macro F1-score, precision, recall, and accuracy, to assess the overall performance of the model.

#### **6.1.8. Impact of Activation Function**

The incorporation of these activation functions within our custom model serves the purpose of enhancing its expressive power. By introducing non-linearity through ReLU, the model can effectively learn intricate patterns and relationships in the data. Moreover, the Sigmoid activation provides a probabilistic interpretation of the model's output, which can be valuable for binary classification tasks.

Throughout our experiments, we have systematically assessed the impact of these activation functions on various performance metrics, including accuracy, precision, recall, and F1-score. This evaluation has enabled us to discern which activation function yields the best overall performance for our specific sequence classification task.

In summary, our investigation into the role of activation functions within our custom model has provided valuable insights into how non-linearity influences the model's capability to capture complex patterns and make precise predictions. These findings contribute to a deeper understanding of the significance of activation functions in neural network architectures and offer essential guidance for designing effective models, particularly in the context of sequence classification tasks.

In conclusion, fine-tuning pre-trained model parameters is a delicate process that requires thoughtful consideration of various factors. Batch size, training epochs, and warm-up ratios can significantly affect both model performance and training efficiency. A balance must be struck to achieve the best trade-offs between these factors while considering the specific requirements and constraints of the research setup.

## **6.2. Key Findings**

RQ1: What are the research gaps in ATC?

Most previous studies have concentrated on surveying methods and procedures. There has, however, been no attempt to provide an overview of ATC research trends and patterns. Understanding current trends and patterns would help researchers and practitioners better direct their efforts toward enhancing language processing and producing more sustainable Arabic-based applications. This motivated us to conduct a thorough literature analysis to find

contemporary ATC trends and patterns, as well as proposed solution types. It is believed that this new review would provide a thorough basis for the state-of-the-art in ATC and advance the research wheel for further ATC studies.

The analysis results showed that studies published on ATC can be classified into three different categories, including topic area, task/ application, and ATC phases. It has been noticed that 46% of the analyzed studies have focused on enhancing and improving ATC phases and processes. Unlike the previous reviews that focused on a specific domain, this review analyzed the ATC published literature through various domains, including healthcare, legal, security and cybersecurity, history, culture, and religion, social media, and agriculture. This taxonomy would help practitioners to understand the state-of-the-art domains and consider the challenges and opportunities when attempting to develop Arabic NLP applications.

While analyzing the existing ATC literature, several challenges were noticed. First, there is no clear open-source corpus. Second, the dataset preprocessing steps are not easy due to the ambiguity of the language. Third, the number of publications in this field is still insufficient to reach the English text classification. This, in turn, opens a large room for further developments in ATC. Fourth, most of the developed models were task specific. There is a need for a more generalized approach that can be applied regardless of the task. Fifth, most of the analyzed studies concentrated on improving one or two phases of the preprocessing stage. Further research might consider this gap and attempt to develop an approach that enhances all the phases. Sixth, the focus of most of the studies was to improve the accuracy and processing time. However, no attempt was made to study the cost of the classification process.

Also, several noteworthy methodologies have emerged for further investigation. Among these, BERT, a model known for its cutting-edge performance in Natural Language Processing (NLP), holds great promise. The exploration of ensemble models, integrating BERT with neural networks, presents an avenue for potentially yielding enhanced outcomes. Moreover, the inherent benefits of BERT itself offer valuable contributions to the field.

It is evident from the SLR that most studies have primarily focused on refining approaches and classifiers, with limited attention given to enhancing the preprocessing stage. Addressing this gap is critical, particularly in exploring novel preprocessing strategies. For instance, substituting emojis with their corresponding words instead of eliminating them represents an intriguing approach worth considering.

Another significant gap identified in the ATC literature is the issue of unbalanced datasets. To tackle this limitation, various solutions have been proposed, including the use of Class-Balanced (CB) SoftMax loss, Random Under-Sampling (RUS), and Random Over-Sampling (ROS). However, further research in this area is warranted to develop more effective approaches in handling unbalanced datasets.

In summary, the systematic literature review highlights the need for more comprehensive investigations into the potential of BERT and ensemble models in ATC, as well as the significance of improving preprocessing strategies and addressing the challenges posed by unbalanced datasets in text classification.

RQ2: What are the effects of utilizing pre-processing methods on the performance of Arabic Text Classification?

Utilizing pre-processing methods in Arabic text classification has a significant impact on the performance of the classification task. This section focuses on the effects of three pre-processing submodules: AraBERT preprocessing, Emoji-to-word interpretation, and Punctuation Removal.

AraBERT preprocessing is specifically designed for the Arabic language and involves several tasks, including text cleaning, normalization, tokenization, and diacritization. By applying these steps, AraBERT preprocessing addresses the challenges posed by Arabic text, such as complex morphology and the presence of diacritics. This preprocessing significantly improves the accuracy and performance of machine learning and NLP models trained on Arabic text data. It ensures standardized and transformed text, leading to better feature representation and more effective classification.

The Emoji-to-word interpretation algorithm introduces a novel approach to handle emojis in Arabic offensive content classification. Emojis play a crucial role in conveying emotions and attitudes within text, providing valuable contextual information. Instead of removing emojis, this algorithm substitutes them with their corresponding literal meanings. By preserving the informative value of emojis, this preprocessing technique enhances the accuracy and comprehension of offensive language identification systems. Previous studies have acknowledged the importance of emojis in offensive language identification, and this algorithm leverages their contextual information effectively. It ensures that the presence or absence of emojis is considered as a key feature in the classification process, contributing to improved classification accuracy.

Punctuation Removal, although already included in the AraBERT preprocessing, requires an additional step to address punctuation marks that remain after emoji interpretation. This step ensures that all punctuation marks are appropriately translated or removed. By removing punctuation, the preprocessed text becomes cleaner and more suitable for subsequent classification tasks. It helps eliminate potential noise and unnecessary variations in the text, allowing the model to focus on meaningful features and patterns, thus improving the classification performance.

The combined effects of these pre-processing methods significantly impact the performance of Arabic text classification. AraBERT preprocessing standardizes and transforms the text, ensuring its suitability for machine learning and NLP tasks. The Emoji-to-word interpretation algorithm preserves the contextual information of emojis, enhancing offensive language identification. Punctuation Removal eliminates noise and unnecessary variations, providing a cleaner input for classification models. These pre-processing steps collectively improve the accuracy, reliability, and generalizability of the classification models, leading to more effective identification and moderation of offensive content in Arabic text.

In conclusion, the utilization of pre-processing methods in Arabic text classification has substantial effects on the performance of the classification task. AraBERT preprocessing addresses the challenges posed by Arabic text, while the Emoji-to-word interpretation algorithm leverages the valuable contextual information of emojis. Punctuation Removal further enhances the quality of the preprocessed text. By incorporating these pre-processing methods, the accuracy and effectiveness of Arabic text classification models are significantly improved, allowing for more reliable identification and moderation of offensive content.

RQ3: What is the most effective model for improving the accuracy of Arabic offensive text classification using pre-trained models?

The most effective model for improving the accuracy of Arabic offensive text classification using pre-trained models is a custom PyTorch model that combines several components. The proposed model utilizes the AraBERT pre-trained language model as its base and incorporates additional layers to enhance its performance.

The AraBERT model is based on Google's BERT architecture and has been pre-trained on a large Arabic corpus consisting of millions of articles from various news sources. It has undergone fine-tuning on multiple tasks, including sequence classification, named entity recognition, and question answering, resulting in state-of-the-art performance on Arabic dialects.

The custom model architecture consists of the AraBERT model, an LSTM layer, and three linear layers. The LSTM layer captures sequential information in the input sequence, which is particularly relevant for processing text data. The three linear layers, designed as fully connected neural network layers, transform the output of the LSTM layer, and produce the final classification probabilities for the offensive or non-offensive classes.

During training, the model's parameters are fine-tuned using the AdamW optimizer and a learning rate scheduler. The model is trained for a specified number of epochs, utilizing a batch size of 16 per device and gradient accumulation steps to reduce memory requirements. Mixed precision training is employed to accelerate the training process on compatible GPUs.

To evaluate the model's performance, a 5-fold cross-validation approach is used. The dataset is divided into five equal parts, and the model is trained on four parts while using the remaining part as the validation set. This process is repeated five times to ensure that each part of the dataset is used for validation exactly once. The model checkpoints are saved based on the F1-macro score, and the best model is selected at the end of training.

The macro F1-score is used as the metric for selecting the best model, as it balances the F1-scores of both offensive and non-offensive classes. Early stopping is employed to prevent overfitting, and the tokenizer specific to the model architecture is used for text tokenization.

Overall, the proposed custom model, combining AraBERT, LSTM, and linear layers, has shown to be highly effective in improving the accuracy of Arabic offensive text classification. The model's performance is enhanced through fine-tuning on a target dataset and the incorporation of sequential information capturing layers. The training process utilizes cross-validation, optimizer, learning rate scheduler, and early stopping techniques to optimize the model's performance and prevent overfitting.

RQ4: To what extent does the proposed model be able to classify Arabic offensive text?

The proposed model demonstrates a high level of effectiveness in classifying Arabic offensive text. The performance evaluation of the model, as presented in Table 6 and Table 7, clearly shows its capability in accurately capturing offensive content in the given task.

The evaluation metrics used, including accuracy and F1-macro score, provide a comprehensive assessment of the model's performance. The proposed model outperforms

various other classifier models and achieves state-of-the-art results. The baseline AraBERT model, which serves as a reference point, achieves an accuracy of 0.82, while the proposed model achieves a significantly higher accuracy of 0.87. This substantial improvement highlights the effectiveness of the proposed model in accurately classifying Arabic offensive text.

Table 20 further supports the exceptional performance of the proposed model by comparing it to previous studies conducted on the same dataset. The results clearly indicate that the proposed model surpasses all other models, establishing its superiority in terms of accuracy and F1-score. For instance, the proposed model achieves an accuracy of 0.87, while the previous best-performing model achieved an accuracy of 0.864.

The evaluation is not limited to a single dataset. The proposed model is also evaluated on the semEval dataset, where it achieves an impressive accuracy of 0.93 and an F1-score of 0.90. This demonstrates the generalizability and effectiveness of the proposed model across different datasets, further solidifying its ability to classify Arabic offensive text.

The success of the proposed model can be attributed to several key factors. Firstly, the utilization of the AraBERT pre-trained language model as the base model provides a strong foundation for understanding Arabic text and capturing offensive content. AraBERT has been fine-tuned on a large corpus and has achieved state-of-the-art performance on Arabic dialects.

Additionally, the incorporation of an LSTM layer and three linear layers in the model architecture enables the capture of sequential information and the mapping of the pre-trained model's output to the text classification task. These added layers enhance the model's ability to capture complex relationships in the data and improve accuracy.

Moreover, the preprocessing techniques employed, such as interpreting emojis with words, removing Arabic stop words, and removing punctuation, contribute to the model's performance. These techniques improve the accuracy and F1-score by effectively preparing the text data for classification.

In conclusion, the proposed model exhibits a high degree of effectiveness in classifying Arabic offensive text. Its superior performance, as demonstrated through the evaluation metrics, outperforms other models, and achieves state-of-the-art results. The combination of the AraBERT pre-trained model, LSTM layer, linear layers, and preprocessing techniques contributes to its success. The model's ability to accurately classify offensive text is evident across different datasets, indicating its generalizability and robustness.

### **6.3. Theoretical Contributions**

The theoretical contributions of this research make significant advancements in the field of Arabic text classification, addressing the specific challenges posed by the Arabic language. The study introduces novel methodologies, algorithms, and models that effectively handle Arabic text, considering its unique linguistic characteristics such as morphology, syntax, and semantic ambiguity. These methodological innovations provide a solid foundation for accurate and efficient Arabic text classification, contributing to the development of more robust models.

Furthermore, this research explores innovative approaches for classification in Arabic text. By investigating linguistic patterns, leveraging domain-specific knowledge, and utilizing pre-trained language models specifically tailored for Arabic, the study enhances the ability to capture relevant features and representations specific to Arabic language data. This in-depth

exploration leads to improved accuracy and effectiveness in the classification process, pushing the boundaries of current methodologies.

Moreover, the study investigates the generalizability and scalability of the proposed approach by evaluating its performance across diverse data sources. By testing the model in various real-world scenarios, the research provides insights into the adaptability of the proposed approach. This analysis enhances the practical utility of the model, demonstrating its potential to be applied to different contexts and domains.

Additionally, this research contributes to the theoretical foundations of Arabic text classification by extending or refining existing theoretical frameworks and concepts. By building upon or challenging previous theories, the study enriches the understanding of the underlying principles and mechanisms involved in Arabic text classification. This contribution expands the theoretical knowledge base, enabling future researchers to build upon this work and deepen their understanding of the field.

Furthermore, the thesis conducts a comprehensive comparative analysis to evaluate the proposed approach against existing methods and frameworks in Arabic text classification. By highlighting the advantages, limitations, and distinguishing factors of the proposed approach, the research provides valuable insights into its unique contributions and differentiating features. This analysis not only demonstrates the novelty and effectiveness of the proposed approach but also facilitates informed decision-making for researchers and practitioners in choosing the most suitable method for Arabic text classification tasks.

In summary, the theoretical contributions of this research encompass methodological innovations, advancements in classification algorithms, exploration of generalizability and

scalability, contributions to theoretical frameworks, and a comprehensive comparative analysis. These contributions collectively enhance the theoretical understanding of Arabic text classification and establish a strong foundation for the development of more accurate and effective classification models in practical applications.

#### **6.4. Practical Implication**

The practical implications of this research extend to various domains and stakeholders, encompassing decision makers, developers, and policy makers. Decision makers in organizations, ranging from government agencies to private companies, can utilize the research findings to make informed choices about incorporating Arabic text classification systems into their operations. By understanding the performance and effectiveness of the proposed approach or model, decision makers can assess its potential impact on enhancing processes such as information retrieval, content filtering, or sentiment analysis in Arabic text.

For example, in the field of customer service, decision makers can deploy the research outcomes to improve automated response systems for Arabic-speaking customers. By implementing the proposed approach, they can enhance the accuracy of classifying customer queries and direct them to the appropriate departments or resources, leading to more efficient customer support experiences.

Developers, on the other hand, can benefit from this research by integrating the proposed approach into their existing text classification systems. This integration can significantly improve the accuracy and reliability of the system in handling Arabic language data. Developers can leverage the research findings and methodologies to enhance their

systems' performance in various applications, such as automatic tagging, sentiment analysis, or content recommendation.

Furthermore, developers can customize and optimize the approach based on their specific requirements and system constraints. They can fine-tune the model by incorporating additional domain-specific knowledge or refining the algorithm to cater to their specific use cases. This flexibility allows developers to create practical implementations of Arabic text classification that align with their applications or platforms, ultimately enhancing user experiences and system functionalities.

Policy makers also play a crucial role in utilizing the research findings to shape guidelines and regulations in the context of Arabic text classification. As the use of text classification extends to areas such as content moderation, hate speech detection, or misinformation identification, policy makers can employ the insights gained from this research to draft policies that address these challenges effectively.

By considering the strengths and limitations of the proposed approach, policy makers can create frameworks that promote ethical and responsible use of Arabic text classification technologies. They can establish guidelines for deploying these systems in a way that safeguards users' rights, protects against potential biases or discriminatory outcomes, and fosters a safe online environment for Arabic-speaking communities.

Furthermore, policy makers can leverage the research findings to support legislative efforts addressing emerging challenges in the digital realm. For instance, they can use the insights gained from the comparative analysis to inform discussions on regulating automated

decision-making systems or shaping policies related to data privacy and protection in Arabic text classification applications.

In summary, the practical implications of this research offer decision makers the opportunity to make informed choices, developers the ability to enhance their systems' performance, and policy makers the means to establish robust frameworks. These implications collectively contribute to advancing the practical implementation and utilization of Arabic text classification, benefiting various stakeholders in real-life contexts. By harnessing the research findings, stakeholders can effectively leverage Arabic text classification technologies to improve decision-making processes, enhance user experiences, and promote responsible and ethical practices in a rapidly evolving digital landscape.

## **6.5. Discussion of Challenges Faced**

During this research, several challenges were encountered. One notable challenge was the issue of data imbalance, where the offensive class was significantly smaller in size than the non-offensive class. Balancing the dataset through oversampling or under sampling techniques introduced complexities and potential biases into the model. Additionally, handling Arabic text posed tokenization and preprocessing challenges, as Arabic has unique linguistic characteristics.

## **6.6. Implications for the NLP Field**

The findings of this research have significant implications for the NLP field. They demonstrate the effectiveness of advanced models like AraBERT and LSTM with pre-trained embeddings in Arabic text classification tasks. This knowledge can inform the development of

more accurate and efficient content moderation systems and sentiment analysis tools for Arabic content, contributing to safer online environments and improved user experiences.

### **6.7. Comparison with Previous Research**

Our results show competitive or superior performance compared to prior research. While some earlier studies have used traditional classifiers and basic text preprocessing, our approach, incorporating AraBERT, NN and LSTM with pre-trained embeddings, achieved higher accuracy and F1-scores. This indicates that leveraging deep learning techniques and pre-trained language models can lead to advancements in Arabic text classification.

### **6.8. Sensitivity Analysis**

Multiple runs with different random seeds were performed to ensure the robustness of our model. These sensitivity analyses yielded consistent results, reaffirming the reliability and stability of our findings.

### **6.9. Addressing Limitations**

It is essential to acknowledge the limitations of this study. One limitation is that the experiments were conducted on a specific set of datasets, and the generalizability of the model to other domains or languages may vary. Additionally, while sensitivity analyses were performed, they may not account for all possible variations in real-world data.

### **6.10. Resource Allocation**

The reproduction of this study required significant computational resources, including GPU clusters and substantial training time. Preprocessing and fine-tuning AraBERT, as well

as experimenting with various neural network architectures, demanded substantial development effort and computational power.

## 6.11. Error Analysis

To gain insights into model failures, an error analysis was conducted. The model occasionally misclassified tweets containing sarcasm or colloquial language as offensive. It also struggled with identifying offensive content embedded within non-offensive context. These findings suggest that contextual understanding remains a challenge for the model.

Error analysis involves examining cases where a machine learning model makes incorrect predictions compared to the actual ground truth. In this context, it appears that the error analysis is focused on cases where the model incorrectly classified text as either offensive (OFF) or not offensive (NOT\_OFF). The table provided contains examples of such errors, including the text, the predicted class by the model, and the actual class.

|    | Text                                                                                                                                                                                                                    | Predicted Class | Actual Class |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|--------------|
| 1  | 😊👉[مستخدم] افطرت عليك بقاء واثنين من فروخها الجن                                                                                                                                                                        | OFF             | NOT_OFF      |
| 2  | 😊داليا _ مبارك مادري ليش تقرفت #                                                                                                                                                                                        | OFF             | NOT_OFF      |
| 5  | URL 🙏اكيد اخس شي                                                                                                                                                                                                        | OFF             | NOT_OFF      |
| 18 | الاشكال هذي اللي تطلع 📺📺أبن _ مدعج _ الوابلي _ ربع _ مليون # سعودي _ الاصل _ و _ المنشا #<br>وللاسف الاعلام السعودي نايم ومكبر الوساده ومخليهم على راحتهم يمثلون الوطن 😊بالاعلام ونقول انا سعودي<br>URL 📺📺بشكل غير صحيح | OFF             | NOT_OFF      |

Here's a breakdown of the error analysis for the provided examples:

1. Example 1:

- Text: "مستخدم افطرت عليك بعقاء واثنين من فروخها الجن" 🍴😊"
- Predicted Class: OFF (Offensive)
- Actual Class: NOT\_OFF (Not Offensive)
- Error Type: False Positive
- Analysis: In this case, the model incorrectly classified a text that appears to be humorous and non-offensive as offensive. This is a false positive error, where the model is overly sensitive to offensive content.

2. Example 2:

- Text: "داليا \_ مبارك مادري ليش تقرفت #" 😬"
- Predicted Class: OFF (Offensive)
- Actual Class: NOT\_OFF (Not Offensive)
- Error Type: False Positive
- Analysis: The model misclassified a text containing an emoji that might imply disgust as offensive. It appears to be a case where the model misinterpreted the context and classified it incorrectly.

3. Example 5:

- Text: "اكيد اخس شي" 🤖 URL"
- Predicted Class: OFF (Offensive)
- Actual Class: NOT\_OFF (Not Offensive)

- Error Type: False Positive
- Analysis: The model again predicted an offensive label for a text containing an emoji indicating frustration. This is another case where the model's sensitivity to certain emojis led to a false positive.

#### 4. Example 18:

- Text: "أبن \_ مدعج \_ الوائلي \_ ربع \_ مليون # \_ سعودي \_ الاصل \_ و \_ المنشا #  
وللاسف الاعلام السعودي نايم ومكبر الوساده ☹️ الاشكال هذي اللي تطلع بالاعلام وتقول انا سعودي  
ومخليهم على راحتهم يمثلون الوطن بشكل غير صحيح URL"
- Predicted Class: OFF (Offensive)
- Actual Class: NOT\_OFF (Not Offensive)
- Error Type: False Positive
- Analysis: The model wrongly classified a text that appears to contain a critique of media portrayal as offensive. This demonstrates the model's inability to understand the context and intent of the text.

In summary, the error analysis reveals that the model tends to produce false positives by incorrectly classifying texts containing certain emojis or contextual nuances as offensive when they are not. This suggests that the model may need further training or fine-tuning to improve its understanding of context and reduce such errors. Additionally, reviewing these errors can help in refining the model's features or data preprocessing to enhance its performance in classifying offensive content accurately.

Suggestions for addressing false positive errors in classifying offensive content:

1. **Data Augmentation:** Enhance the diversity of the training dataset, encompassing a broader spectrum of examples that encompass emojis and nuanced language. This augmentation aids in furnishing the model with a richer understanding of the contextual usage of these linguistic elements.
2. **Contextual Embeddings:** Leverage state-of-the-art pre-trained contextual embeddings, such as BERT, RoBERTa, or GPT-3. These embeddings are adept at capturing contextual nuances within text, which is particularly advantageous for comprehending the connotations and sentiments conveyed by different emojis and expressions.
3. **Ensemble Models:** Employ other ensemble modeling techniques to amalgamate the predictions of multiple models. This ensemble approach mitigates the impact of false positives emanating from a single model, often resulting in enhanced overall performance.
4. **Threshold Adjustment:** Experiment with the calibration of the classification threshold. Tailoring the threshold to distinguish text as offensive affords control over the trade-off between precision and recall, allowing for the reduction of false positives while accounting for potential increases in false negatives.
5. **Post-processing Rules:** Institute post-processing rules to filter out false positives. These rules can encompass contextual checks for specific emojis and expressions, thereby refining the model's predictions based on contextual cues.
6. **Active learning and Human Review:** Integrate a human review component into the classification pipeline. Following the model's predictions, human reviewers can

evaluate borderline cases where the distinction between offensive and non-offensive content is less clear. This reviewer feedback serves the dual purpose of model refinement and augmenting the quality of training data. Implement active learning methodologies wherein the model identifies uncertain cases and solicits human input for labeling. This iterative approach empowers the model to gradually improve its performance, particularly on challenging instances.

7. **Regular Model Retraining:** Institute periodic model retraining to accommodate evolving data. Offensive language and contextual nuances evolve over time; hence, recurrently updating the model with current data ensures its adaptability to shifting linguistic trends.
8. **Feedback Mechanism:** If feasible, introduce a user feedback mechanism that allows users to report false positives. This feedback loop contributes valuable data for model refinement and continuous improvement.
9. **Explainability:** Employ explainable AI techniques to gain insights into the rationale behind the model's predictions. This analytical approach aids in identifying specific features or patterns contributing to false positives, thereby guiding model enhancement efforts.

These strategies, when judiciously applied either in isolation or in combination, offer effective avenues to mitigate the issue of false positive errors in the classification of offensive content. It is imperative to strike a balance between reducing false positives and upholding high classification accuracy, with the selection of specific strategies contingent on the characteristics of the dataset and the application context.

## **6.12. Conclusion**

In conclusion, this chapter has discussed the results in the context of the research question, addressed challenges faced, highlighted implications for the NLP field, theoretical and practical implication, conducted sensitivity analyses, addressed limitations, discussed resource allocation, and performed an error analysis. The outcomes of this research provide valuable insights into the potential of advanced NLP techniques for Arabic text classification and pave the way for future improvements and applications in content moderation and sentiment analysis.

## Chapter 7: Conclusion

Furthermore, the chapter acknowledges and discusses the limitations of the work. It critically evaluates the constraints, challenges, or shortcomings that were encountered during the research process. This may include issues related to data availability, sample size, algorithmic limitations, or any other factors that may have influenced the results. Addressing the limitations provides insights for future research directions and helps in understanding the scope and boundaries of the current study.

Overall, this chapter serves as a summary of the key findings, explains their significance, and discusses the limitations that should be considered when interpreting the results.

### 7.1. Main Findings

RQ1: What are the research gaps in Arabic Text Classification (ATC)?

- Previous studies in ATC mainly focused on surveying methods and procedures, lacking an overview of ATC research trends and patterns.
- ATC research can be categorized into three main areas: topic area, task/application, and ATC phases.
- Challenges in ATC include the absence of a clear open-source corpus, dataset preprocessing complexity, and the need for more publications in this field.
- Most ATC models are task-specific, and there's a need for a generalized approach. Research should also focus on improving all phases of preprocessing and understanding the cost of classification.

RQ2: What are the effects of utilizing pre-processing methods on the performance of Arabic Text Classification?

- Pre-processing methods significantly impact the performance of Arabic Text Classification.
- AraBERT preprocessing addresses Arabic language challenges, improving accuracy and NLP model performance.
- Emoji-to-word interpretation preserves contextual information, enhancing offensive language identification.
- Punctuation Removal ensures cleaner input, reducing noise and improving classification performance.

RQ3: What is the most effective model for improving the accuracy of Arabic offensive text classification using pre-trained models?

- The most effective model combines AraBERT, LSTM, and linear layers.
- AraBERT provides a strong foundation for understanding Arabic text.
- The LSTM layer captures sequential information, enhancing model performance.
- Fine-tuning, cross-validation, and early stopping techniques optimize model accuracy.

RQ4: To what extent does the proposed model be able to classify Arabic offensive text?

- The proposed model demonstrates high effectiveness in classifying Arabic offensive text.
- It outperforms other classifiers and achieves state-of-the-art results, with an accuracy of 0.87.
- Generalizability is confirmed by achieving an accuracy of 0.93 on the semEval dataset.
- The model's success is attributed to AraBERT, LSTM, linear layers, and effective preprocessing techniques.

Overall, the findings highlight the potential for improving ATC, the impact of preprocessing methods, and the effectiveness of a custom model for Arabic offensive text classification.

## **7.2. Contribution of the Research**

Our research contributes significantly to the field of ATC in several ways:

- **Comprehensive Overview:** We provide a comprehensive overview of ATC research trends and patterns, enabling researchers and practitioners to better direct their efforts in the development of Arabic NLP applications. This overview fills a critical gap in the existing literature.
- **Preprocessing Insights:** Our novel demojized algorithm enhances the effectiveness of preprocessing techniques. The techniques identified can be directly applied to enhance classification accuracy.
- **Effective Model:** Our proposed model, incorporating AraBERT, LSTM, and linear layers, establishes a new benchmark for ATC accuracy. It presents a

valuable contribution to the field, offering a strong foundation for future research.

### **7.3. Limitations and Future Work**

Overall, the proposed framework for improving the accuracy of Arabic offensive dataset classification showed promising results, but there are some limitations that should be considered. One limitation of the proposed framework is the imbalanced nature of the dataset used for training and evaluation. Imbalanced datasets, where one class is significantly more dominant than others, can pose challenges in accurately capturing the minority class. In the context of offensive dataset classification, this imbalance can result in a model that is more biased towards the majority class, leading to lower accuracy in identifying offensive instances. Future research could explore alternative methods to address this issue, such as oversampling techniques for the minority class, under sampling techniques for the majority class, or the use of advanced algorithms specifically designed to handle imbalanced datasets. Moreover, researchers could consider collecting a more balanced dataset that adequately represents the distribution of offensive and non-offensive language in real-world scenarios.

Another limitation pertains to the potential bias present in the dataset. The offensive language used in the dataset may not fully encompass the diverse range of offensive language used in the Arabic language. This can result in a model that is biased towards the specific instances of offensive language present in the training data, potentially leading to inaccuracies when faced with new or different offensive language patterns. To address this limitation, future research could explore strategies to mitigate bias in the dataset. This could involve collecting a more diverse dataset that represents a broader range of offensive language

expressions, considering different dialects or socio-cultural contexts. Additionally, techniques like data augmentation could be employed to generate more diverse examples of offensive language, thereby reducing potential bias, and improving the model's ability to generalize to different offensive language patterns.

Furthermore, it is worth considering the generalizability of the proposed framework beyond the specific dataset and context used in the study. While the framework demonstrated promising results in the context of Arabic offensive dataset classification, its effectiveness in different domains, platforms, or languages may vary.

Future work could focus on evaluating the framework's performance in diverse settings and investigating any necessary adaptations or modifications required to ensure its generalizability.

In conclusion, while the proposed framework for improving the accuracy of Arabic offensive dataset classification has shown promise, there are several limitations that need to be addressed. These include the imbalanced nature of the dataset and potential bias in the offensive language representation. Future research can explore methods to mitigate these limitations, such as employing techniques to balance the dataset, collecting more diverse and representative data, and evaluating the framework's generalizability. By addressing these limitations, further advancements can be made in the field of Arabic offensive dataset classification, leading to more accurate and reliable models for identifying and mitigating offensive language in Arabic text.

Based on the findings and discussions presented in this study, the following recommendations are proposed for future research:

- Explore alternative preprocessing techniques: Although the proposed model achieved impressive results, further investigation into alternative preprocessing techniques is warranted. This includes exploring different methods of handling emojis, stop words, and punctuation, as well as considering the impact of these techniques on model performance. Additionally, investigating the use of other language-specific preprocessing tools and techniques may lead to further improvements in accuracy.
- Expand the evaluation to diverse datasets: To assess the generalizability of the proposed model, future research should conduct evaluations on diverse datasets from various domains, including social media, news articles, and online forums. This will help validate the effectiveness of the model across different contexts and provide insights into its robustness and adaptability.
- Consider domain-specific offensive content: Offensive content varies across different domains, such as politics, religion, or gender. Future research should investigate domain-specific offensive language detection and classification, as it requires a deeper understanding of the specific context and nuances associated with each domain. Developing domain-specific models can enhance accuracy and relevance in detecting offensive content within specific domains.
- Evaluate interpretability and bias: It is crucial to evaluate the interpretability of the proposed model and address any potential bias it may exhibit. Future research should focus on techniques that provide insights into the model's decision-making process,

allowing for better understanding and identification of biases. Additionally, exploring ways to mitigate bias and enhance fairness in offensive content classification models is an important direction for future research.

- Explore cross-lingual transfer learning: Arabic offensive content classification can benefit from cross-lingual transfer learning approaches. Investigating the transferability of models trained on other languages, such as English, to Arabic offensive content classification can be an interesting avenue for future research. This would involve adapting and finetuning pre-existing models to the Arabic language, thereby leveraging the knowledge gained from other languages.

## **7.4. Conclusion**

This study focused on improving the accuracy of Arabic offensive dataset classification using pre-trained models and preprocessing techniques. Through extensive experiments and analysis, we arrived at several key findings.

Firstly, we proposed an ensemble model that combines a finetuned AraBERT with two NN layers, LSTM, and ReLU and Sigmoid activation functions. This model achieved the state-of-the-art performance, surpassing all other approaches in terms of accuracy and F1-score. These results highlight the effectiveness and potential of leveraging advanced models like AraBERT in the detection and classification of offensive language in Arabic text. The proposed model demonstrated practicality and suitability for addressing offensive content detection and moderation in Arabic language-based platforms and applications.

Regarding preprocessing techniques, we found that the AraBERT processor was essential in improving accuracy. However, the inclusion of additional steps such as removing

stop words and the most frequent words did not yield further accuracy improvements. This suggests that while preprocessing can enhance accuracy, it is crucial to carefully consider the specific techniques used and their impact on the dataset. In investigating the impact of different approaches to handling emojis, we found that translating emojis to their corresponding Arabic meaning achieved a higher accuracy rate compared to prior studies. This approach showed promise in improving the classification of offensive Arabic text by capturing the nuanced aspects of offensive language. However, our final proposed model revealed a slight decrease in accuracy when replacing emojis, indicating the potential loss of important contextual information. This highlights the complexity of dealing with emojis in Arabic offensive content classification and the need for more tailored approaches. Overall, our study addressed the research questions set forth by identifying effective strategies for improving accuracy in Arabic offensive dataset classification and evaluating the impact of preprocessing techniques and emoji handling. The proposed model and findings contribute to advancement of the field of offensive content detection in Arabic text.

In conclusion, the exceptional performance of our proposed model, the ensemble of a finetuned AraBERT with LSTM, NN layers and various activation functions, highlights its superiority in accurately capturing offensive content in Arabic. The combination of domain expertise, experimental exploration, and understanding of the target dataset and available resources is crucial for achieving improved accuracy in classifying Arabic offensive datasets using pre-trained models. While the AraBERT processor is a valuable preprocessing technique, the inclusion of additional steps should be carefully considered to avoid potential negative effects on accuracy. The translation of emojis to their corresponding Arabic meaning showed promise in improving classification accuracy, but further research is needed to address

the potential loss of contextual information. Our study provides insights and a strong foundation for future research in Arabic offensive content classification and the development of effective models and preprocessing techniques.

## References

- Abandah, G. A. (2015). Automatic diacritization of Arabic text using recurrent neural networks. *Document Analysis and Recognition*, vol. 18(2), pp. 83–197.
- Abdeen, M. A. R. (2019). A closer look at arabic text classification,” *International Journal of Advanced Computer Science and Applications. International Journal of Advanced Computer Science and Applications*, vol. 10(11), pp. 677–688.
- Abdeen, M. A. R., AlBouq, S., Elmahalawy, A. & Shehata, S. (2019). A closer look at arabic text classification. *International Journal of Advanced Computer Science and Applications*, vol. 10(11), pp. 677–688.
- Abdelaal, H. M., Elmahdy, A. N., Halawa, A. A. & Youness, H. A. (2018). Improve the automatic classification accuracy for Arabic tweets using ensemble methods. *Journal of Electrical Systems and Information Technology*. Electronics Research Institute (ERI), (2017), pp. 1–8.
- Abdelgwad, M. M., Soliman, T. H. A. & Taloba, A. I. (2022). Arabic aspect sentiment polarity classification using BERT. *Journal of Big Data*. Springer International Publishing, vol. 9(1).
- Abdul-Mageed, M., Elmadany, A. R. & Nagoudi, E. M. B. (2021). ARBERT & MARBERT: Deep bidirectional transformers for Arabic. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, (i), pp. 7088–7105.
- Abdulaziz, S., Abdul, A. & Jilani, K. (2020). Machine learning approach for threat detection

- on social media posts containing Arabic text. *Evolutionary Intelligence*. Springer Berlin Heidelberg, (0123456789).
- Abooraig, R., Al-zu'bi, S., Kanan, T., Hawashin, B., Al Ayoub, M. & Hmeidi, I. (2018). Automatic categorization of Arabic articles based on their political orientation Automatic categorization of Arabic articles based on their political orientation. *Digital Investigation*. Elsevier Ltd, (December), pp. 1–18.
- Abu-farha, I. & Magdy, W. (2020). From Arabic Sentiment Analysis to Sarcasm Detection : The ArSarcasm Dataset. *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*. European Language Resource Association, pp. 32–39 [online]. Available at: <https://aclanthology.org/2020.osact-1.5>.
- Abuzeina, D. & Al-anzi, F. S. (2017). Employing fisher discriminant analysis for Arabic text classification. *Computers and Electrical Engineering*. Elsevier Ltd, vol. 0, pp. 1–13.
- Al\_Janabi, S., Salman, M. A. & Mohammad, M. (2019). Multi-level Network Construction Based on Intelligent Big Data Analysis. *Studies in Big Data*. Springer Science and Business Media Deutschland GmbH, vol. 53, pp. 102–118.
- Al\_Janabi, S., Salman, M. A. & Mohammed, M. (2020). Pragmatic Text Mining Method to Find the Topics of Citation Network. *Lecture Notes in Networks and Systems*. Springer, vol. 81, pp. 190–205.
- Al-anzi, F. S. & Abuzeina, D. (2017). Towards an Enhanced Arabic Text Classification Using Cosine Similarity and Latent Semantic Indexing. *JOURNAL OF KING SAUD UNIVERSITY - COMPUTER AND INFORMATION SCIENCES*. King Saud University, pp. 189–195.

- Al-anzi, F. S. & Abuzeina, D. (2018). Beyond vector space model for hierarchical Arabic text classification : A Markov chain approach. *Information Processing and Management*. Elsevier, vol. 54(1), pp. 105–115.
- Al-badarneh, A., Al-Shawakfa, E., Bani-Ismail, B., Al-Rababah, K. & Shatnawi, S. (2017). The impact of indexing approaches on Arabic text classification. *Journal of Information Science*, vol. 43(2), pp. 159–173.
- Al-Emran, M., Zaza, S. & Shaalan, K. (2015). Parsing modern standard Arabic using Treebank resources. *2015 International Conference on Information and Communication Technology Research, ICTRC 2015*.
- Al-ghamdi, S., Al-khalifa, H. & Al-Salman, A. (2023). Fine-Tuning BERT-Based Pre-Trained Models for Arabic Dependency Parsing. *Applied Sciences*, vol. 13(7), p. 4225.
- Al-Janabi, S. (2021). Overcoming the Main Challenges of Knowledge Discovery through Tendency to the Intelligent Data Analysis. *2021 International Conference on Data Analytics for Business and Industry, ICDABI 2021*. Institute of Electrical and Electronics Engineers Inc., pp. 286–294.
- Al-Janabi, S., Alkaim, A. F. & Adel, Z. (2020). An Innovative synthesis of deep learning techniques (DCapsNet & DCOM) for generation electrical renewable energy from wind energy. *Soft Computing*. Springer, vol. 24(14), pp. 10943–10962.
- Al-Janabi, S., Mohammad, M. & Al-Sultan, A. (2020). A new method for prediction of air pollution based on intelligent computation. *Soft Computing*. Springer, vol. 24(1), pp. 661–680.
- Al-Janabi, S., Patel, A., Fatlawi, H., Kalajdzic, K. & Al Shourbaji, I. (2015). Empirical rapid and accurate prediction model for data mining tasks in cloud computing environments.

- 2014 *International Congress on Technology, Communication and Knowledge, ICTCK 2014*. Institute of Electrical and Electronics Engineers Inc.
- Al-Jarf, R. (2023). Non-conventional spelling in informal, colloquial arabic writing on Facebook. *International Journal of Linguistics, Literature and Translation*, vol. 6.4, pp. 35–47.
- Al-Radaideh, Q. A. & Al-Abrat, M. A. (2019). An Arabic text categorization approach using term weighting and multiple reducts. *Soft Computing*. Springer Berlin Heidelberg, vol. 23(14), pp. 5849–5863.
- Al-Saleh, A. B. & Menai, M. E. B. (2016). Automatic Arabic text summarization: a survey. *Artificial Intelligence Review*. Springer Netherlands, vol. 45, pp. 203–234.
- Al-sarem, M., Emara, A. H. & Wahab, A. (2020). Performance of Authorship Attribution Classifiers with Short Texts Application of Religious Arabic Fatwas. *International Journal of Data Mining, Modelling and Management*, (January), pp. 350–364.
- AL-Sarem, M. & Emara2, A.-H. (2019). The effect of training set size in authorship attribution : application on short Arabic texts. *International Journal of Electrical and Computer Engineering*, vol. 9(1), pp. 652–659.
- Al-shaibani, M. S., Alyafeai, Z. & Ahmad, I. (2020). Meter classification of Arabic poems using deep bidirectional recurrent neural networks. *Pattern Recognition Letters, ELSEVIER*. Elsevier B.V., vol. 136, pp. 1–7.
- Al-Smadi, M. (2019). Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews. *International Journal of Machine Learning and Cybernetics*, vol. 10(8), pp. 2163–2175.
- Al-Tamimi, A.-K., Bani-Isaa, E. & Al-Alami, A. (2021). Active Learning for Arabic Text

- Classification. *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pp. 123–126.
- Alabbas, W. (2016). Arabic text classification methods: Systematic literature review of primary studies. *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pp. 361–367.
- Alabbas, W., Al-Khateeb, H. M. & Mansour, A. (2016). Arabic Text Classification Methods : Systematic Literature Review of Primary Studies. *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*. IEEE, pp. 361–367.
- Alahmadi, A., Joorabchi, A. & Mahdi, A. E. (2017). Combining Words and Concepts for Automatic Arabic. *International Conference on Arabic Language Processing*. Springer, Cham, pp. 105–119.
- Alammary, A. S. (2022). BERT Models for Arabic Text Classification: A Systematic Review. *Applied Sciences (Switzerland)*, vol. 12(11).
- Alanazi, S. A. (2019). Towards Identifying Features for Automatic Gender Detection : A Corpus Creation and Analysis. *IEEE Access*. IEEE, vol. 7, pp. 111931–111943.
- Alayba, A. M., Palade, V., England, M. & Iqbal, R. (2017). Arabic Language Sentiment Analysis on Health Services. *2017 IEEE International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pp. 114–118.
- Alghamdi, N. (2019). Monitoring Mental Health Using Smart Devices with Text Analytical Tool. *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, pp. 2046–2051.
- Alhaj, Y. A., Al-qaness, M., Udara, W., Hussain, A. & Abdelaal, H. (2018). Efficient Feature Representation Based on the Effect of Words Frequency for Arabic Documents

- Classification. *Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering*. acm, pp. 397–401.
- Alharthi, R., Alhothali, A. & Moria, K. (2021a). A real-time deep-learning approach for filtering Arabic low-quality content and accounts on Twitter. *Information Systems* 99, p. 101740.
- Alharthi, R., Alhothali, A. & Moria, K. (2021b). A real-time deep-learning approach for filtering Arabic low-quality content and accounts on Twitter. *Information Systems*, vol. 99, p. 101740.
- Alhashmi, S. F. S. (2020). Systematic Review of the Factors Affecting the Artificial Intelligence Implementation in the Health Care. *The International Conference on Artificial Intelligence and Computer Vision*, pp. 37–49.
- Alhawarat, M. & Aseeri, A. O. (2020). A Superior Arabic Text Categorization Deep Model (SATCDM). *IEEE Access*, vol. 8, pp. 24653–24661.
- Alhozaimi, A. & Almishari, M. (2018). Arabic Twitter Profiling For Arabic-Speaking Users. *2018 21st Saudi Computer Society National Computer Conference (NCC)*. IEEE, pp. 1–6.
- Alhumoud, S. O. & Al Wazrah, A. A. (2022). Arabic sentiment analysis using recurrent neural networks: a review. *Artificial Intelligence Review*. Springer, vol. 55, pp. 707–748.
- Aljedani, N., Alotaibi, R. & Taileb, M. (2020). HMATC : Hierarchical multi-label Arabic text classification model using machine learning. *Egyptian Informatics Journal*. Faculty of Computers and Information, Cairo University, (xxxx).
- AlKhamissi, B. & Diab, M. (2022). Meta AI at Arabic Hate Speech 2022: MultiTask Learning with Self-Correction for Hate Speech Classification. *5th Workshop Open-Source*

*Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation Conference, LREC 2022*, pp. 186–193.

Alkhatib, M., El Barachi, M. & Shaalan, K. (2019). An Arabic social media based framework for incidents and events monitoring in smart cities. *Journal of Cleaner Production*.

Elsevier Ltd, vol. 220, pp. 771–785.

Alkhatib, M., El, M. & Shaalan, K. (2019). An Arabic Social Media Based Framework for Incidents and Events Monitoring in Smart Cities An Arabic social media based framework for incidents and events monitoring in smart cities. *Journal of Cleaner Production*. Elsevier Ltd, vol. 220(March), pp. 771–785.

Almaliki, M., Almars, A. M., Gad, I. & Atlam, E. S. (2023). ABMM: Arabic BERT-Mini Model for Hate-Speech Detection on Social Media. *Electronics (Switzerland)*, vol. 12(4), pp. 1–16.

Almuzaini, H. A. & Azmi, A. M. (2020). Impact of Stemming and Word Embedding on Deep Learning-Based Arabic Text Categorization. *IEEE Access*, vol. 8, pp. 127913–127928.

Alnemer, K. A., Alhuzaim, W. M., Alnemer, A. A. & Alharbi, B. B. (2015). Are Health-Related Tweets Evidence Based ? Review and Analysis of Health-Related Tweets on Twitter. *JOURNAL OF MEDICAL INTERNET RESEARCH*, vol. 17, pp. 1–6.

Alorini, D. & Rawat, D. (2019). Automatic Spam Detection on Gulf Dialectical Arabic Tweets. *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, pp. 448–452.

Alqudsi, A., Omar, N. & Shaker, K. (2014). Arabic machine translation: a survey. *Artificial Intelligence Review*. Springer, vol. 42, pp. 549–572.

- Alruily, M., Fazal, A. M. & Mostafa, A. M. (2023). Automated Arabic Long-Tweet Classification Using Transfer Learning with BERT. *Applied Sciences (Switzerland)*, vol. 13(6).
- AlSaleh, D. & Larabi-Marie-Sainte, S. (2021). Arabic Text Classification using Convolutional Neural Network and Genetic Algorithms. *IEEE Access*. IEEE, vol. 9, pp. 91670–91685.
- Alshaer, H. N., Otair, M. A. & Abualigah, L. (2021). Feature selection method using improved CHI Square on Arabic text classifiers: analysis and application. *Multimedia Tools and Applications*, vol. 80, pp. 10373–10390.
- Alshalabi, H., Tiun, S., Omar, N., AL-Aswadi, F. N. & Ali Alezabi, K. (2021). Arabic Light-based Stemmer Using New Rules. *Journal of King Saud University - Computer and Information Sciences*. The Authors.
- Altamimi, M. & Teahan, W. J. (2019). Arabic Dialect Identification of Twitter Text Using PPM Compression. *International Journal of Computational Linguistics (IJCL)*, vol. 10(4), pp. 47–59.
- Alwaneen, T. H., Azmi, A. M., Aboalsamh, H. A., Cambria, E. & Hussain, A. (2022). Arabic question answering system: a survey. *Artificial Intelligence Review*. Springer, vol. 55, pp. 207–253.
- Alyafeai, Z., Al-shaibani, M. S., Ghaleb, M. & Ahmad, I. (2021). Evaluating Various Tokenizers for Arabic Text Classification. *arXiv preprint arXiv:2106.07540*, vol. 5 [online]. Available at: <http://arxiv.org/abs/2106.07540>.
- Alzu'bi, S., Ferreira, T. C., Pavanelli, L. & Al-Badrashiny, M. (2022). aiXplain at Arabic Hate Speech 2022: An Ensemble Based Approach to Detecting Offensive Tweets. *5th*

- Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation Conference, LREC 2022*, pp. 214–217.
- Ameur, M. S. H. & Aliane, H. (2021). AraCOVID19-MFH: Arabic COVID-19 Multi-label Fake News & Hate Speech Detection Dataset. *Procedia Computer Science*. Elsevier B.V., vol. 189, pp. 232–241.
- Antoun, W., Baly, F. & Hajj, H. (2020). Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.
- Antoun, W., Baly, F. & Hajj, H. (2023). AraGPT2: Pre-Trained Transformer for Arabic Language Generation. *In Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pp. 196–207 [online]. Available at: <https://aclanthology.org/2021.wanlp-1.21>.
- Atwan, J., Wedyan, M., Bsoul, Q., Hamadeen, A., Alturki, R. & Ikram, M. (2021). The Effect of using Light Stemming for Arabic Text Classification. *International Journal of Advanced Computer Science and Applications*, vol. 12(5), pp. 768–773.
- Awal, M. R., Lee, R. K. W., Tanwar, E., Garg, T., & Chakraborty, T. (2023). Model-agnostic meta-learning for multilingual hate speech detection. *Systems. IEEE Transactions on Computational Social*.
- Ayed, R., Labidi, M. & Maraoui, M. (2017). Arabic Text Classification : New study. 2017 *International Conference on Engineering & MIS (ICEMIS)*. IEEE, pp. 1–7.
- Badaro, G., Baly, R., Hajj, H., El-Hajj, W., Shaban, K. B., Habash, N., Al-Sallab, A. & Hamdi, A. (2019). A survey of opinion mining in Arabic: A comprehensive system perspective covering challenges and advances in tools, resources, models, applications,

- and visualizations. *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 18(3).
- Bahassine, S., Madani, A. & Kissi, M. (2017). A new similarity measure for automatic text categorization based on vector space model. *Proceedings of the Second International Conference on Advanced Wireless Information, Data, and Communication Technologies*, pp. 1–7.
- Bekkali, M. & Lachkar, A. (2017). Web Search Engine-Based Representation for Arabic Tweets Categorization. *Social Data Mining and Analysis to Prediction and Community Detection*, pp. 79–101.
- Chantar, H., Mafarja, M., Alsawalqah, H., Asghar, A., Aljarah, I. & Aljarah, I. (2019). Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. *Neural Computing and Applications*. Springer London, vol. 6.
- Chouigui, A., Khiroun, O. Ben & Elayeb, B. (2017). ANT Corpus : An Arabic News Text Collection for Textual Classification. *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. IEEE.
- Cohen, K. B. (2004). Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization. *Language*, vol. 80(1), p. 178.
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L. & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451.
- D. Bahdanau, K.H. Cho, Y. B. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473*.

- D. Svozil. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems.*, vol. 39(1), pp. 43–63.
- Daif, M., Kitada, S. & Iyatomi, H. (2020). AraDIC: Arabic Document Classification Using Image-Based Character Embeddings and Class-Balanced Loss. *The 58th Annual Meeting of the Association for Computational Linguistics*, pp. 214–221.
- Darwish, K., Habash, N., Abbas, M., Al-Khalifa, H., Al-Natsheh, H. T., Bouamor, H., Bouzoubaa, K., Cavalli-Sforza, V., El-Beltagy, S. R., El-Hajj, W., Jarrar, M. & Mubarak, H. (2021). A panoramic survey of natural language processing in the Arab world. *Communications of the ACM*, vol. 64(4), pp. 72–81.
- Devlin, J., Chang, M. W., Lee, K. & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1(Mlm), pp. 4171–4186.
- Dürscheid, C. and Siever, C. M. (2017). Beyond the alphabet–communcataion of emojis. *Kurzfassung eines (auf Deutsch) zur Publikation eingereichten Manuskripts.*
- El-Alami, F.-Z. & El Alaoui, S. (2018). Word Sense Representation based-method for Arabic Text Categorization. *2018 9th International Symposium on Signal, Image, Video and Communications (ISIVC)*. IEEE, pp. 141–146.
- El-Alami, F., El Mahdaouy, A., El Alaoui, S. O. & En-Nahnahi, N. (2020a). A deep autoencoder-based representation for Arabic text categorization. *Journal of Information and Communication Technology*, vol. 19(3), pp. 381–398.
- El-Alami, F. Z., El Alaoui, S. O. & Nahnahi, N. E. (2021). A multilingual offensive language

- detection method based on transfer learning from transformer fine-tuning model. *Journal of King Saud University-Computer and Information Sciences*. The Authors.
- El-Alami, F. Z., El Mahdaouy, A., El Alaoui, S. O. & En-Nahnahi, N. (2020b). A deep autoencoder-based representation for Arabic text categorization. *Journal of Information and Communication Technology*, vol. 19(3), pp. 381–398.
- El-Alami, F. zahra, Ouatik El Alaoui, S. & En Nahnahi, N. (2021a). A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model. *Journal of King Saud University - Computer and Information Sciences*. The Authors.
- El-Alami, F. zahra, Ouatik El Alaoui, S. & En Nahnahi, N. (2021b). Contextual semantic embeddings based on fine-tuned AraBERT model for Arabic text multi-class categorization. *Journal of King Saud University - Computer and Information Sciences*. The Authors.
- El-Halees, A. M. (2017). Arabic Opinion Mining Using Distributed Representations of Documents. *Proceedings - 2017 Palestinian International Conference on Information and Communication Technology, PICICT 2017*, pp. 28–33.
- Elayeb, B. (2019). Arabic word sense disambiguation: a review. *Artificial Intelligence Review*. Springer, vol. 52, pp. 2475–2532.
- Elhassan, R. & Ahmed, M. (2015). Arabic Text Classification review, vol. 4(1), pp. 1–5.
- Elkaref, N. & Abu-Elkheir, M. (2022). GUCT at Arabic Hate Speech 2022: Towards a Better Isotropy for Hate Speech Detection. *5th Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation*

- Conference, LREC 2022*, pp. 209–213.
- Elnagar, A., Al-Debsi, R. & Einea, O. (2020). Arabic text classification using deep learning models. *Information Processing and Management*. Elsevier, vol. 57(1), p. 102121.
- Farghaly, A. & Shaalan, K. (2009). Arabic Natural Language Processing: Challenges and Solutions. *ACM Transactions on Asian Language Information Processing*, vol. 8, pp. 1–10.
- Farha, I. A. & Magdy, W. (2021). Benchmarking Transformer-based Language Models for Arabic Sentiment and Sarcasm Detection. *Proceedings of the Sixth Arabic Natural Language Processing Workshop*.
- Faris, H., Habib, M., Faris, M., Alomari, A., Castillo, P. A. & Alomari, M. (2021). Classification of Arabic healthcare questions based on word embeddings learned from massive consultations: a deep learning approach. *Journal of Ambient Intelligence and Humanized Computing*. Springer Berlin Heidelberg.
- Farokhian, M., Rafe, V. & Veisi, H. (2022). Fake news detection using parallel BERT deep neural networks. *arXiv preprint arXiv:2204.04793* [online]. Available at: <http://arxiv.org/abs/2204.04793>.
- Ferrag, M. A., Friha, O., Kantarci, B., Tihanyi, N., Cordeiro, L., Debbah, M., ... & Choo, K. K. R. (2023). Edge Learning for 6G-enabled Internet of Things: A Comprehensive Survey of Vulnerabilities, Datasets, and Defenses. *arXiv preprint arXiv:2306.10309*.
- Gaussier, E. & Goutte, C. (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. *Proceedings of the 27th European conference on Advances in Information Retrieval Research*.
- Ghourabi, A., Mahmood, M. A. & Alzubi, Q. (2020). A Hybrid CNN-LSTM Model for SMS

- Spam Detection in Arabic and English Messages. *Future Internet*, vol. 12(9), p. 156.
- Gouiouez, M. & Hadni, M. (2017). Multi-Agent System for Arabic Text Categorization. *Proceedings of the Mediterranean Symposium on Smart City Applications*. Springer, Cham, pp. 161–174.
- Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure [online]. Available at: <http://arxiv.org/abs/2203.05794>.
- Guru, D. S., Ali, M. & Suhil, M. (2018). A Novel Term Weighting Scheme and an Approach for Classification of Agricultural Arabic Text Complaints. *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*. IEEE, pp. 24–28.
- Hadj Ameur, M. S. & Aliane, H. (2021). AraCOVID19-MFH: Arabic COVID-19 Multi-label Fake News & Hate Speech Detection Dataset. *Procedia CIRP*. Elsevier B.V., vol. 189, pp. 232–241.
- Hadni, M. & Gouiouez, M. (2017). Graph Based Representation for Arabic Text Categorization. *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*. acm, pp. 1–7.
- Harbecke, D., Chen, Y., Hennig, L. & Alt, C. (2022). Why only Micro-F1? Class Weighting of Measures for Relation Classification. *NLP-Power 2022 - 1st Workshop on Efficient Benchmarking in NLP, Proceedings of the Workshop*, pp. 32–41.
- Harrag, F. (2009a). Neural Network for Arabic Text Classification. *Second International Conference on the Applications of Digital Information and Web Technologies*, pp. 778–783.
- Harrag, F. (2009b). Neural Network for Arabic Text Classification, pp. 778–783.

Hassan, F., Satori, K., Yahyaouy, A., El Moubtahij, H. & Lamtougui, H. (2020).

APPLICATIONS OF DEEP LEARNING IN ARABIC SENTIMENT ANALYSIS :  
RESEARCH PERSPECTIVE. In *2020 1st International Conference on Innovative  
Research in Applied Science, Engineering and Technology (IRASET)*. IEEE, pp. 20–  
25.

Hegazi, M. O., Al-Dossari, Y., Al-Yahy, A., Al-Sumari, A. & Hilal, A. (2021). Preprocessing  
Arabic text on social media. *Heliyon*. Elsevier Ltd, vol. 7(2), p. e06191.

Hijazi, M., Zeki, A. & Ismail, A. (2021). Arabic Text Classification Using Hybrid Feature  
Selection Method Using Chi-Square Binary Artificial Bee Colony Algorithm.  
*International Journal of Mathematics and Computer Science*, vol. 16(1), pp. 213–228.

Hochreiter, S. & Schmidhuber, J. (1997a). Long Short-Term Memory. *Neural Computation*,  
vol. 9(8), pp. 1735–1780.

Hochreiter, S. & Schmidhuber, J. U. (1997b). Long shortterm memory. *Neural Computation*,  
vol. 9(8), pp. 1735–1780.

Howard, J. & Ruder, S. (2018). Universal language model fine-tuning for text classification.  
*ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics,  
Proceedings of the Conference (Long Papers)*, vol. 1, pp. 328–339.

Hriez, S. & Awajan, A. (2020). Authorship Identification for Arabic Texts Using Logistic  
Model Tree Classification. *Science and Information Conference*. Springer International  
Publishing, pp. 656–666.

Hussein, S., Farouk, M. & Hemayed, E. (2019). Gender identification of egyptian dialect in  
twitter. *Egyptian Informatics Journal*. Faculty of Computers and Information, Cairo  
University, vol. 20(2), pp. 109–116.

- Ikram, A. Y. & Chakir, L. (2019). Arabic Text Classification in the Legal Domain. *2019 3rd International Conference on Intelligent Computing in Data Sciences, ICDS 2019*. IEEE, pp. 1–6.
- Jarrar, M., Khalilia, M. & Ghanem, S. (2022). Wojood: Nested Arabic Named Entity Corpus and Recognition using BERT. *2022 Language Resources and Evaluation Conference, LREC 2022*, (June), pp. 3626–3636.
- Khedher, M. I., Jmila, H. & El-yacoubi, M. A. (2020). Automatic processing of Historical Arabic Documents : A comprehensive Survey Automatic processing of Historical Arabic Documents : A comprehensive Survey. *Pattern Recognition, Elsevier*. Elsevier Ltd, vol. 100(November), p. 107144.
- Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 253–258.
- Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J. & Linkman, S. (2009). Systematic literature reviews in software engineering - A systematic literature review. *Information and Software Technology*. Elsevier B.V., vol. 51(1), pp. 7–15.
- Koshiry, A. M. El, I., Entesar Hamed, E., Abd El-Hafeez, T. & Omar, A. (2023). BERT-based Approach to Arabic Hate Speech and Offensive Language Detection in Twitter: Exploiting Emojis and Sentiment Analysis. *Big Data and Cognitive Computing*.
- Larabi Marie-Sainte, S. & Alalyani, N. (2020). Firefly Algorithm based Feature Selection for Arabic Text Classification. *Journal of King Saud University - Computer and Information Sciences*. King Saud University, vol. 32(3), pp. 320–328.
- Lee, D. (2017). Long short-term memory recurrent neural network-based acoustic model using

- connectionist temporal classification on a large-scale training corpus. *China Communication*, vol. 14(9), pp. 23–31.
- Li, J., Wang, B. & Ding, H. (2020). Lijunyi at SemEval-2020 Task 4: An ALBERT Model based Maximum Ensemble with Different Training Sizes and Depths for Commonsense Validation and Explanation. *14th International Workshops on Semantic Evaluation, SemEval 2020 - co-located 28th International Conference on Computational Linguistics, COLING 2020, Proceedings*, pp. 556–561.
- Li, P., Guo, F. & Shang, J. (2020). User-Guided Aspect Classification for Domain-Specific Texts. *arXiv preprint arXiv:2004.14555*.
- Lippincott, T., Mcnamee, P. & Duh, K. (2019). JHU System Description for the MADAR Arabic Dialect Identification Shared Task. *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pp. 264–268.
- M, H. & M.N, S. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, vol. 5(2), pp. 01–11.
- Madhfar, M. A. H. & Al-Hagery, M. A. H. (2019). Arabic text classification: A comparative approach using a big dataset. *2019 International Conference on Computer and Information Sciences, ICCIS 2019. IEEE*, (4), pp. 1–5.
- Mahdi, M. A. & Al\_Janabi, S. (2020). A Novel Software to Improve Healthcare Base on Predictive Analytics and Mobile Services for Cloud Data Centers. *Lecture Notes in Networks and Systems*. Springer, vol. 81, pp. 320–339.
- Makram, K. H., Nessim, K. G., Abd-Almalak, M. E., Roshdy, S. Z., Salem, S. H., Thabet, F. & Mohamed, E. H. (2022). CHILLAX - at Arabic Hate Speech 2022: A Hybrid

- Machine Learning and Transformers based Model to Detect Arabic Offensive and Hate Speech. *5th Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation Conference, LREC 2022*, pp. 194–199.
- Marie-Sainte, S. L. & Alalyani, N. (2020). Firefly Algorithm based Feature Selection for Arabic Text Classification. *Journal of King Saud University - Computer and Information Sciences*. King Saud University, vol. 32(3), pp. 320–328.
- Mei, Q. (2019). Decoding the new world language: Analyzing the popularity, roles, and utility of emojis. *In Companion Proceedings of The 2019 World Wide Web Conference*, pp. 417–418.
- Mohammad, A. H. & Al-momani, O. (2016). Arabic Text Categorization Using Support vector machine , Naïve Bayes and Neural Network. *Research Gate*.
- Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., Altman, D., Antes, G., Atkins, D., Barbour, V., Barrowman, N., Berlin, J. A., Clark, J., Clarke, M., Cook, D., D'Amico, R., Deeks, J. J., Devereaux, P. J., Dickersin, K., Egger, M., Ernst, E., Gøtzsche, P. C., Grimshaw, J., Guyatt, G., Higgins, J., Ioannidis, J. P. A., Kleijnen, J., Lang, T., Magrini, N., McNamee, D., Moja, L., Mulrow, C., Napoli, M., Oxman, A., Pham, B., Rennie, D., Sampson, M., Schulz, K. F., Shekelle, P. G., Tovey, D. & Tugwell, P. (2009). Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Medicine*.
- Mostafa, A., Mohamed, O. & Ashraf, A. (2022). GOF at Arabic Hate Speech 2022: Breaking The Loss Function Convention For Data-Imbalanced Arabic Offensive Text Detection.

- 5th Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation Conference, LREC 2022*, pp. 167–175.
- Mubarak, H., Hassan, S. & Chowdhury, S. A. (2022a). Emojis as Anchors to Detect Arabic Offensive Language and Hate Speech, pp. 1–21.
- Mubarak, H., Hassan, S. & Chowdhury, S. A. (2022b). Emojis as Anchors to Detect Arabic Offensive Language and Hate Speech. *arXiv preprint arXiv:2201.06723*, pp. 1–21.
- Mubarak, H., Rashed, A., Darwish, K., Samih, Y. & Abdelali, A. (2021). Arabic Offensive Language on Twitter: Analysis and Experiments. *WANLP 2021 - 6th Arabic Natural Language Processing Workshop, Proceedings of the Workshop*, pp. 126–135.
- Nessir, M. A. Ben, Rhouma, M., Haddad, H. & Fourati, C. (2022). iCompass at Arabic Hate Speech 2022: Detect Hate Speech Using QRNN and Transformers. *5th Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation Conference, LREC 2022*, pp. 176–180.
- Omar, A., Mahmoud, T. M., Abd-El-Hafeez, T. & Mahfouz, A. (2021). Multi-label Arabic text classification in Online Social Networks. *Information Systems*. Elsevier Ltd., vol. 100, p. 101785.
- Oo, S. H. P., Theeramunkong, T. & Hung, N. D. (2020). Sentence Sentiment Classification Using Convolutional Neural Network in Myanmar Texts. *Proceedings of the 2020 2nd International Conference on Image, Video and Signal Processing*, pp. 144–149.
- de Paula, A. F. M., Rosso, P., Bensalem, I. & Zaghouani, W. (2022). UPV at the Arabic Hate Speech 2022 Shared Task: Offensive Language and Hate Speech Detection using

- Transformers and Ensemble Models. *5th Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation Conference, LREC 2022*, pp. 181–185.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, vol. 21, pp. 1–67.
- Rayson, P. & Alsudiasl, L. (2019). Classifying Information Sources in Arabic Twitter to Support Online Monitoring of Infectious Diseases. *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics: (WACL-3)*. Association for Computational Linguistics. [online]. Available at: <https://www.aclweb.org/anthology/W19-5604>.
- Rumelhart, D. E. (1986). Learning representations by back-propagating errors. *Nature*, vol. 323(6088), pp. 533–536.
- S. A. Salloum. (2020). Mining in Educational Data: Review and Future Directions. *The International Conference on Artificial Intelligence and Computer Vision*, pp. 92–102.
- Safaya, A., Abdullatif, M. & Yuret, D. (2020). KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media. *14th International Workshops on Semantic Evaluation, SemEval 2020 - co-located 28th International Conference on Computational Linguistics, COLING 2020, Proceedings*, pp. 2054–2059.
- Sainath, T. N. (2013). Deep Convolutional Neural Networks for LVCSR. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8614–8618.
- Salama, R. A., Youssef, A. & Fahmy, A. (2018). Morphological Word Embedding for Arabic.

- Procedia Computer Science*. Elsevier B.V., vol. 142, pp. 83–93.
- Salloum, S. A. (2020a). A Survey of Semantic Analysis Approaches. *The International Conference on Artificial Intelligence and Computer Vision*, pp. 61–70.
- Salloum, S. A. (2020b). Machine Learning and Deep Learning Techniques for Cybersecurity: A Review. *The International Conference on Artificial Intelligence and Computer Vision*, pp. 50–57, 2.
- Salloum, S. A., Al-Emran, M., Monem, A. A. & Shaalan, K. (2018). *Using text mining techniques for extracting information from research articles. Studies in Computational Intelligence*.
- Salloum, S. A., Mhamdi, C., Al-Emran, M. & Shaalan, K. (2017). Analysis and Classification of Arabic Newspapers' Facebook Pages using Text Mining Techniques. *International Journal of Information Technology and Language Studies*, vol. 1(2), pp. 8–17.
- Santos, M. L. B. dos. (2022). The “so-called” UGC: an updated definition of user-generated content in the age of social media. *Online Information Review* 46, vol. 1, pp. 95–113.
- Shannag, F., Hammo, B. H. & Faris, H. (2022). *The design, construction and evaluation of annotated Arabic cyberbullying corpus. Education and Information Technologies*. Springer US.
- Shapiro, A., Khalafallah, A. & Torki, M. (2022). AlexU-AIC at Arabic Hate Speech 2022: Contrast to Classify. *5th Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation Conference, LREC 2022*, pp. 200–208.
- Strubell, E. (2019). Machine Learning Models for Efficient and Robust Natural Language

Processing, (October).

- Sundus, K., Al-Haj, F. & Hammo, B. (2019). A Deep learning approach for Arabic text classification. *2019 2nd International Conference on New Trends in Computing Sciences, ICTCS 2019 - Proceedings*. IEEE, pp. 1–7.
- Sutskever, I. (2011). Generating text with recurrent neural networks. *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 1017-1024,.
- Touahri, I., & Mazroui, A. (2021). Enhancement of a multi-dialectal sentiment analysis system by the detection of the implied sarcastic features. *Knowledge-Based Systems*, vol. 227.
- Undavia, S., Meyers, A. & Ortega, J. E. (2018). A comparative study of classifying legal documents with neural networks. *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, FedCSIS 2018*, vol. 15, pp. 515–522.
- Vaswani, A., Shazeer, N., Pham, H., Parmar, N. & Uszkoreit, J. (2017). Attention is all you need. *Adv. Neural Inf. Process. Sys.*
- Wahdan, A., Al-Emran, M. & Shaalan, K. (2023). A systematic review of Arabic text classification : areas , applications , and future directions. *Soft Computing*. Springer Berlin Heidelberg, vol. 6.
- Wahdan, A., Hantoobi, S., Salloum, S. A. & Shaalan, K. (2020a). A systematic review of text classification research based on deep learning models in Arabic language. *International Journal of Electrical and Computer Engineering (IJECE)*, pp. 6629–6643.
- Wahdan, A., Hantoobi, S., Salloum, S. A. & Shaalan, K. (2020b). A systematic review of text classification research based on deep learning models in Arabic language A systematic review of text classification research based on deep learning models in Arabic

- language. *International Journal of Electrical and Computer Engineering (IJECE)*, (December), pp. 6629–6643.
- Wahdan, A., Salloum, S. A. & Shaalan, K. (2021). Text Classification of Arabic Text: Deep Learning in ANLP. *Advances in Intelligent Systems and Computing*, vol. 1339, pp. 95–103.
- Wahdan, A., Salloum, S. A. & Shaalan, K. (2022). Qualitative Study in Natural Language Processing: Text Classification. *Lecture Notes in Networks and Systems*, vol. 322, pp. 83–92.
- Zahidi, Y., El Younoussi, Y. & Azroumahli, C. (2019). Comparative Study of the Most Useful Arabic-supporting Natural Language Processing and Deep Learning Libraries. *2019 International Conference on Optimization and Applications, ICOA 2019*. IEEE, pp. 1–10.
- Zaza, S. & Al-Emran, M. (2015). Mining and exploration of credit cards data in UAE. *Proceedings - 2015 5th International Conference on e-Learning, ECONF 2015*. IEEE, pp. 275–279.
- Zeroual, I., Goldhahn, D., Eckart, T. and Lakhouaja, A. (2019). OSIAN: Open Source International Arabic News Corpus - Preparation and Integration into the CLARIN-infrastructure. *In Proceedings of the fourth arabic natural language processing workshop*, pp. 175–182.
- Zhong, B. (2019). Convolutional neural network : Deep learning-based classification of building quality problems. *Advanced Engineering Informatics*, vol. 40, pp. 46–57.

## Appendices

### Appendix A

The below tables show the emoji dictionary created by the algorithm. It shows each emoji with its corresponding name in English and Arabic

| emoji | emoji name                       | emoji_name_ar                |    |                                |                                   |
|-------|----------------------------------|------------------------------|----|--------------------------------|-----------------------------------|
| 😉     | :winking_face:                   | :الوجه الغطس :               | 🍝  | :spaghetti:                    | :معكرونة:                         |
| 💧     | :pear:                           | :كُنْزَى:                    | 😓  | :grinning_face_with_sweat:     | :الوجه الابتسامة مع العرق:        |
| 💩     | :pile_of_poo:                    | :كومة من البراز :            | 👒  | :woman's_hat:                  | :قبعة المرأة:                     |
| 📦     | :wrapped_gift:                   | :هدية ملفوفة :               | 💖  | :sparkling_heart:              | :القلب الفوار:                    |
| 🐝     | :honeybee:                       | :عسل النحل:                  | I  | I                              | I                                 |
| o     | o                                | o                            | 😐  | :neutral_face:                 | :وجه محايد:                       |
| 🌐     | :globe_with_meridians:           | :Globe مع Meridians:         | R  | R                              | R                                 |
| 🤔     | :thinking_face:                  | :وجه التفكير :               | @  | @                              | @                                 |
| 🌳     | :tanabata_tree:                  | :Tanabatare:                 | 🌵  | :cactus:                       | :صبار:                            |
| 🎆     | :sparkler:                       | :Sparkler:                   | —  | —                              | -                                 |
| 👐     | :wind_face:                      | :وجه الرياح :                | 👐  | :palms_up_together:            | :أشجار النخيل معا:                |
| 👤     | :man:                            | :رجل:                        | 👠  | :high-heeled_shoe:             | :حذاء بكعب عال:                   |
| 😊     | :smiling_face_with_smiling_eyes: | :الوجه المبتسم بعيون مبتسمة: | 🔊  | :speaker_high_volume:          | :مكبر صوت كبير:                   |
| 🍡     | :pine_decoration:                | :ديكور الصنوبر :             | 🐸  | :frog:                         | :ضفدع:                            |
| G     | G                                | G                            | 💄  | :lipstick:                     | :أحمر الشفاه:                     |
| 😂     | :face_with_tears_of_joy:         | :وجه بدموع الفرح :           | "  | "                              | "                                 |
| 😊     | :slightly_smiling_face:          | :الوجه المبتسم قليلاً :      | 👉  | :backhand_index_pointing_left: | :مؤشر الظهر الذي يشير إلى اليسار: |
| ✉️    | :incoming_envelope:              | :الظرف الوارد :              | ✎  | :pencil:                       | :قلم:                             |
| 🖐️    | :vulcan_salute:                  | :تحية فولكان :               | ☕  | :hot_beverage:                 | :المشروبات الساخنة:               |
| 👼     | :baby_angel:                     | :الطفل الملاك:               | ☀️ | :sun:                          | :شمس:                             |

|    |                                  |                                    |                                                                                   |                            |                               |
|----|----------------------------------|------------------------------------|-----------------------------------------------------------------------------------|----------------------------|-------------------------------|
| □  | :lizard:                         | :سحلية:                            |  | :minibus:                  | :حافلة صغيرة :                |
| 😊  | :relieved_face:                  | :الوجه المرتاح :                   | 👻                                                                                 | :ghost:                    | :شبح:                         |
| □  | :smiling_face_with_open_hands:   | :الوجه المبتسم بأيدي :<br>مفتوحة:  |  | :hamburger:                | :هامبرغر :                    |
| 🚒  | :fire_engine:                    | :محرك ناري:                        | 😱                                                                                 | :face_screaming_in_fear:   | :وجه يصرخ في خوف :            |
| 🏹  | :bow_and_arrow:                  | :القوس والسهام :                   |  | :water_buffalo:            | :جاموس الماء:                 |
| 🍉  | :watermelon:                     | :بطيخ:                             | 💭                                                                                 | :thought_balloon:          | :بالون الفكر :                |
| □  | :scorpion:                       | :برج العقرب:                       | □                                                                                 | :spoon:                    | :ملقعة:                       |
| s  | s                                | s                                  | □                                                                                 | :lion:                     | :أسد:                         |
| 😬  | :pouting_face:                   | :وجه العيوس :                      | 🌑                                                                                 | :new_moon_face:            | :وجه القمر الجديد :           |
| 🙄  | :winking_face_with_tongue:       | :غطس الوجه مع اللسان :             | 👑                                                                                 | :crown:                    | :تاج:                         |
| 🕌  | :kaaba:                          | :الكعبة:                           | ⚖️                                                                                | :balance_scale:            | :مقياس التوازن :              |
| ✊  | ✊                                | ✊                                  | ✊                                                                                 | :raised_fist:              | :رفع قبضة:                    |
| □  | :baguette_bread:                 | :خبز الرغيف الفرنسي :              | 🐘                                                                                 | :boar:                     | :الخنزير :                    |
| 🎵  | :musical_note:                   | :نوتة موسيقية:                     | 🙋                                                                                 | :person_raising_hand:      | :شخص يرفع اليد :              |
| 🖋️ | :fountain_pen:                   | :قلم حبر:                          | ↘️                                                                                | :right_arrow_curving_down: | :منحني السهم الأيمن :         |
| 💃  | :woman_dancing:                  | :امراة ترقص :                      | □                                                                                 | :zany_face:                | :الوجه zany:                  |
| ↗️ | :up-right_arrow:                 | :السهم الصحيح :                    | 🐛                                                                                 | :ant:                      | :النملة :                     |
| 😡  | :anger_symbol:                   | :رمز الغضب :                       | 🗡️                                                                                | :dagger:                   | :خنجر:                        |
| ↖️ | :pick:                           | :يختار:                            | ⚔️                                                                                | :crossed_swords:           | :السيوف المتقاطعة :           |
| ●  | :red_circle:                     | :دائرة حمراء:                      | 😓                                                                                 | :downcast_face_with_sweat: | :الوجه الهبوط مع :<br>العرق:  |
| □  | :chopsticks:                     | :عبدان تناول الطعام :              | 🚁                                                                                 | :helicopter:               | :هليكوبتر:                    |
| ~  | :wavy_dash:                      | :متموج DASH :                      | 📡                                                                                 | :satellite_antenna:        | :هوائي الأقمار :<br>الصناعية: |
| ✋  | :raised_hand:                    | :ترتفع اليد :                      | ◻️                                                                                | :white_small_square:       | :مربع صغير أبيض :             |
| 🍫  | :chocolate_bar:                  | :قطعة شوكولاتة:                    | 🍕                                                                                 | :pizza:                    | :بيتزا:                       |
| 😞  | :disappointed_face:              | :وجه بخيبة أمل :                   | 👁️                                                                                | :eye:                      | :عين:                         |
| □  | :face_with_thermometer:          | :الوجه مع مقياس الحرارة :          | □                                                                                 | :crossed_fingers:          | :تمنيت:                       |
| 😊  | :beaming_face_with_smiling_eyes: | :الوجه المبتهج بعيون :<br>مبتسمون: | 🪲                                                                                 | :lady_beetle:              | :سيدة بيتل :                  |

|                                                                                     |                                 |                                 |                                                                                     |                           |                                  |
|-------------------------------------------------------------------------------------|---------------------------------|---------------------------------|-------------------------------------------------------------------------------------|---------------------------|----------------------------------|
|    | :angry_face_with_horns:         | :وجه غاضب بالقرون :             | ←                                                                                   | left_arrow:               | :السهم الايسر:                   |
|    | :wine_glass:                    | :كأس نبيذ:                      | {                                                                                   | {                         | {                                |
|    | :check_mark_button:             | :زر الاختيار :                  |    | :girl:                    | :بنت:                            |
|    | :rolling_on_the_floor_laughing: | :يتحرج على الارض من :<br>الضحك: |    | :cat_with_tears_of_joy:   | :قطعة مع دموع الفرح :            |
|    | :detective:                     | :المحقق:                        |    | :goat:                    | :معزة:                           |
| -                                                                                   | -                               | -                               |    | :slightly_frowning_face:  | :وجه عيوس قليلا :                |
| ◀                                                                                   | :reverse_button:                | :زر عكسي :                      | □                                                                                   | :money-mouth_face:        | :الوجه الشفهي :                  |
| ✓                                                                                   | :check_mark:                    | :علامة الاختيار:                |    | :grinning_squinting_face: | :وجه التحديق ابتسامة :<br>عريضة: |
|    | :headphone:                     | :سماعة الرأس :                  | □                                                                                   | :fortune_cookie:          | :كعكة الحظ:                      |
| l                                                                                   | ]                               | ]                               |    | :full_moon_face:          | :وجه القمر :                     |
|    | :green_salad:                   | :سلطة خضراء:                    |    | :hushed_face:             | :الوجه الصامت :                  |
|    | :heart_with_ribbon:             | :القلب مع الشريط :              |    | :sleepy_face:             | :وجه نائم:                       |
|   | :coffin:                        | :نعش:                           |   | :rabbit:                  | :أرنب:                           |
|  | :face_vomiting:                 | :القيء الوجه :                  | 🐍                                                                                   | :snake:                   | :تعبان:                          |
|  | :chicken:                       | :فرخة:                          |  | :man's_shoe:              | :حذاء الرجل :                    |
|  | :person_facepalming:            | :شخص الوجه :                    | □                                                                                   | :alarm_clock:             | :منبه:                           |
|  | :koala:                         | :كوالا :                        |  | :thumbs_up:               | :ممتاز:                          |
|  | :people_with_bunny_ears:        | :الناس مع اذان الأرنب :         |  | :sun_with_face:           | :الشمس مع الوجه :                |
|  | :ogre:                          | :غول:                           |  | :wolf:                    | :ذئب:                            |
|  | :meat_on_bone:                  | :اللحوم على العظام :            | ؛                                                                                   | ؛                         | ؛                                |
|  | :revolving_hearts:              | :القلوب الدوارة :               | [                                                                                   | [                         | [                                |
|  | :person_bowing:                 | :شخص ينحني :                    |  | :moai:                    | :جيد لطيف :                      |
|  | :money_bag:                     | :حقيرة المال:                   | □                                                                                   | :butterfly:               | :فراشة:                          |
| »                                                                                   | »                               | »                               |  | :sunflower:               | :دوار الشمس:                     |
|  | :syringe:                       | :حقنة:                          |  | :growing_heart:           | :قلب متزايد :                    |
|  | :hot_springs:                   | :الينابيع الساخنة :             |  | :artist_palette:          | :لوحة الفنانين :                 |
|  | :green_heart:                   | :قلب اخضر:                      |  | :guitar:                  | :غيتار:                          |
| ➡                                                                                   | :crocodile:                     | :تمساح:                         |  | :purse:                   | :حقيبة:                          |

|                                                                                     |                              |                            |                                                                                     |                         |                                   |
|-------------------------------------------------------------------------------------|------------------------------|----------------------------|-------------------------------------------------------------------------------------|-------------------------|-----------------------------------|
|    | :tiger_face:                 | :وجه النمر :               |    | :peach:                 | :خوخ:                             |
|    | :wilted_flower:              | :زهرة ذبل :                |    | :raised_back_of_hand:   | :مرفوع باليد :                    |
|    | :victory_hand:               | :يد النصر :                |    | :scissors:              | :مقص:                             |
|    | :right-facing_fist:          | :قبضة يواجه اليمين :       |    | :medium-dark_skin_tone: | :لون البشرة المتوسطة :<br>:الظلام |
|    | :backhand_index_pointing_up: | :فهرس الظهر يشير إلى :     |    | :see-no-evil_monkey:    | : See-No-Evil<br>Monkey:          |
|    | :cat_face:                   | :وجه القطه:                |    | :bear:                  | :دُب:                             |
| #                                                                                   | #                            | #                          |    | :ox:                    | :ثور:                             |
|    | :round_pushpin:              | :المستدير Pushpin :        |    | :school:                | :مدرسة:                           |
| E                                                                                   | E                            | E                          |    | :tumbler_glass:         | :زجاج البهلوان :                  |
|    | :speak-no-evil_monkey:       | :لا جدوى-قرء الكلام :      |    | :handbag:               | :حقيبة يد:                        |
|    | :heart_with_arrow:           | :قلب مع السهم :            | ^                                                                                   | ^                       | ^                                 |
|    | :microphone:                 | :ميكروفون:                 | '                                                                                   | '                       | '                                 |
|    | :cow_face:                   | :وجه البقرة :              | ‘                                                                                   | ‘                       | ‘                                 |
|   | :door:                       | :باب:                      |   | :ring:                  | :جرس:                             |
|  | :jack-o-lantern:             | : Jack-O-Lantern:          | *                                                                                   | *                       | *                                 |
|  | :sports_medal:               | :الميدالية الرياضية :      |  | :heart_decoration:      | :زخرفة القلب :                    |
| }                                                                                   | }                            | }                          |  | :elephant:              | :الفيل :                          |
| L                                                                                   | L                            | L                          |  | :banana:                | :موز:                             |
| ◆                                                                                   | :small_orange_diamond:       | :الماس البرتقالي الصغير :  |  | :bikini:                | :بيكيني:                          |
| 😊                                                                                   | :face_savoring_food:         | :وجه الطعام المذاق :       | %%                                                                                  | :chains:                | :السلاسل:                         |
| %                                                                                   | %                            | %                          |  | :dolphin:               | :دولفين:                          |
| H                                                                                   | H                            | H                          |                                                                                     |                         |                                   |
| =                                                                                   | =                            | =                          |  | :down-right_arrow:      | :سهم أسفل اليمين :                |
|  | :leopard:                    | :فهد:                      |  | :skull_and_crossbones:  | :جمجمة وعظمتين :<br>:متقاطعتين    |
|  | :teacup_without_handle:      | :بدون مقبض Teacup :        | ♀                                                                                   | :female_sign:           | :علامة أنثى :                     |
|  | :dollar_banknote:            | :الدولار الأوراق النقدية : | *                                                                                   | :eight-spoked_asterisk: | :ثمانى علامات النجمة :            |
|  | :cow:                        | :بقرة:                     |  | :ledger:                | :موازنة:                          |

|   |                                |                               |    |                                  |                                   |
|---|--------------------------------|-------------------------------|----|----------------------------------|-----------------------------------|
| 🔌 | :pill:                         | :حبة:                         | 🔌  | :electric_plug:                  | :المكونات الكهربائية:             |
| ✕ | :cross_mark:                   | :علامة تقاطع:                 | 👩👨 | :woman_and_man_holding_hands:    | :امرأة ورجل يمسكان يديه:          |
| 😡 | :angry_face:                   | :وجه غاضب:                    | 🚓  | :police_car_light:               | :ضوء سيارة الشرطة:                |
| “ | “                              |                               | □  | :deer:                           | :عزيري:                           |
| 😊 | :smiling_face:                 | :الوجه المبتسم:               | 🍂  | :fallen_leaf:                    | :ورقة السقوط:                     |
| ⚡ | :hammer_and_wrench:            | :المطرقة وال مفصورة:          | (  | (                                | ((                                |
| 👷 | :construction_worker:          | :عامل بناء:                   | 👤  | :person_pouting:                 | :شخص العبوس:                      |
| 😩 | :tired_face:                   | :وجه متعب:                    | 💋  | :kissing_face:                   | :التقبيل الوجه:                   |
| 🌱 | :seedling:                     | :الشتلات:                     | 🐷  | :pig_face:                       | :وجه الخنزير:                     |
| 🌙 | :last_quarter_moon_face:       | :الوجه القمر في الربع الأخير: | □  | :pie:                            | :فطيرة:                           |
| ♥ | :heart_suit:                   | :بدلة القلب:                  | 🕷  | :spider:                         | :العنكبوت:                        |
| 🖐 | :hand_with_fingers_splayed:    | :يدوي بأصابع متطفلة:          | 😨  | :fearful_face:                   | :وجه مخيف:                        |
| 🌙 | :first_quarter_moon_face:      | :وجه القمر في الربع الأول:    | 📱  | :mobile_phone:                   | :تليفون محمول:                    |
| 👤 | :person_tipping_hand:          | :شخص يميل اليد:               | □  | :bat:                            | :مضرب:                            |
| 🐳 | :whale:                        | :حوت:                         | 🚲  | :person_biking:                  | :ركوب الدراجات:                   |
| □ | :person_shrugging:             | :الشخص يتجاهل:                | 💋  | :kiss:                           | :قبلة:                            |
| 👋 | :waving_hand:                  | :التلويح باليد:               | 💀  | :skull:                          | :جمجمة:                           |
| B | B                              | B                             | 👣  | :footprints:                     | :اثار الاقدام:                    |
| : | :                              | :                             | 🐉  | :dragon:                         | :التنين:                          |
| □ | :face_with_head-bandage:       | :Bandageالوجه مع :            |    |                                  |                                   |
| ❤ | :beating_heart:                | :قلب نابض:                    | 📺  | :television:                     | :التلفاز:                         |
| 😊 | :smiling_face_with_heart-eyes: | :الوجه المبتسم بعيون القلب:   | □  | :fast_reverse_button:            | :زر عكس سريع:                     |
| Q | Q                              | Q                             | 🍟  | :french_fries:                   | :بطاطس مقلية:                     |
| 💔 | :broken_heart:                 | :قلب مجروح:                   | 🔞  | :no_one_under_eighteen:          | :لا أحد أقل من الثامنة عشر:       |
| 🚩 | :triangular_flag:              | :العلم الثلاثي:               | ☁  | :cloud:                          | :سحاب:                            |
| □ | :face_with_hand_over_mouth:    | :وجه مع يد الفم:              | 🔒  | :locked:                         | :مقفل:                            |
| 😌 | :sad_but_relieved_face:        | :وجه حزين ولكن مرتاح:         | 😺  | :grinning_cat_with_smiling_eyes: | :قطعة ابتسامة عريضة مع عيون تبسم: |

|                                                                                     |                              |                         |                                                                                     |                               |                                     |
|-------------------------------------------------------------------------------------|------------------------------|-------------------------|-------------------------------------------------------------------------------------|-------------------------------|-------------------------------------|
|    | :stadium:                    | :ملعب:                  |    | :face_with_rolling_eyes:      | :وجه بعيون متداولة:                 |
|    | :face_with_symbols_on_mouth: | :وجه مع رموز على الفم:  | γ                                                                                   | γ                             | γ                                   |
|    | :ribbon:                     | :شريط:                  |    | :smiling_face_with_horns:     | :الوجه المبتسم مع قرون:             |
|    | :rooster:                    | :الديك:                 |    | :pineapple:                   | :أناناس:                            |
|    | :high_voltage:               | :الجهد العالي:          |    | :backpack:                    | :حقيبة الظهر:                       |
|    | :face_with_raised_eyebrow:   | :وجه مع الحاجب المرتفع: |    | :nail_polish:                 | :طلاء الأظافر:                      |
|    | :men's_room:                 | :غرفة رجال:             |    | :play_button:                 | :زر التشغيل:                        |
|    | :person_frowning:            | :شخص عبوس:              |    | :person_walking:              | :شخص يمشي:                          |
|    | :trophy:                     | :غنيمة:                 |    | :link:                        | :وصلة:                              |
|    | :left-facing_fist:           | :قبضة يواجه اليسار:     |    | :no_mobile_phones:            | :لا يوجد هواتف:<br>محمولة:          |
|    | :snowman:                    | :الرجل الثلجي:          |    | :jeans:                       | :جينز:                              |
|    | :turkey:                     | :ديك رومي:              |    | :grinning_face_with_big_eyes: | :وجه ابتسامة عريضة:<br>بعيون كبيرة: |
|   | :battery:                    | :بطارية:                |   | :snowflake:                   | :ندفة الثلج:                        |
| D                                                                                   | D                            | D                       |  | :name_badge:                  | :اسم شارة:                          |
|  | :toilet:                     | :الحمام:                |  | :evergreen_tree:              | :شجرة الخضرة:                       |
|  | :books:                      | :كتب:                   |  | :house_with_garden:           | :منزل مع حديقة:                     |
|  | :fire:                       | :نار:                   |  | :umbrella_with_rain_drops:    | :مظلة مع قطرات:<br>المطر:           |
|  | :water_wave:                 | :موجة المياه:           |  | :cyclone:                     | :الإعصار:                           |
|  | :family:                     | :عائلة:                 |  | :person_gesturing_NO:         | :شخص يشير لا:                       |
|  | :persevering_face:           | :الوجه المثابر:         |  | :boy:                         | :ولد:                               |
|  | :raising_hands:              | :رفع اليدين:            | >                                                                                   | >                             | >                                   |
|  | :shushing_face:              | :وجه شوشينغ:            |  | :dog_face:                    | :وجه الكلب:                         |
|  | :input_numbers:              | :أرقام الإدخال:         | κ                                                                                   | κ                             | κ                                   |
|  | :mouth:                      | :فم:                    |  | :glasses:                     | :نظارات:                            |
|  | :bug:                        | :حشرة:                  | +                                                                                   | +                             | +                                   |
|  | :sneezing_face:              | :الوجه العطس:           |  | :clinking_glasses:            | :Clinkingنظارات:                    |
|  | :open_hands:                 | :الأيدي المفتوحة:       | w                                                                                   | w                             | w                                   |

|    |                             |                                   |     |                                 |                               |
|----|-----------------------------|-----------------------------------|-----|---------------------------------|-------------------------------|
| 🍓  | :strawberry:                | :الفراولة:                        | ↖   | :up-left_arrow:                 | :UP-LEFT سهم:                 |
| 🦄  | :unicorn:                   | :وحيد القرن:                      | 👱   | :person_blonde_hair:            | :شعر أشقر شخص:                |
| 👫  | :couple_with_heart:         | :زوجين مع القلب:                  | 🐟   | :fish:                          | :سمكة:                        |
| 💪  | :flexed_biceps:             | :العضلة ذات الرأسين :<br>:المرنة: | 100 | :hundred_points:                | :مائة نقطة:                   |
| 😭  | :loudly_crying_face:        | :وجه البكاء بصوت عالي:            | 👤   | :person_wearing_turban:         | :شخص يرتدي العمامة:           |
| 👉  | :index_pointing_up:         | :فهرس يشير إلى:                   | M   | M                               | M                             |
| 🎾  | :tennis:                    | :تنس:                             | 😬   | :kissing_face_with_closed_eyes: | :تقبيل الوجه بعيون<br>:مغلقة: |
| 👁  | :eyes:                      | :عيون:                            | 😩   | :weary_cat:                     | :قطعة مرهقة:                  |
| 👤  | :face_with_steam_from_nose: | :وجه مع البخار من الأنف:          | 😏   | :smirking_face:                 | :الوجه المبتسم:               |
| ♦  | :diamond_suit:              | :بذلة الماس:                      | 😴   | :sleeping_face:                 | :وجه نائم:                    |
| 🐏  | :ram:                       | :كباش:                            | 😞   | :frowning_face_with_open_mouth: | :وجه عبوس بالفم<br>:المفتوح:  |
| 🚄  | :bullet_train:              | :قطار سريع:                       | 🍀   | :four_leaf_clover:              | :أربع أوراق برسيم:            |
| ☂  | :umbrella:                  | :مظلة:                            | 😓   | :weary_face:                    | :وجه مرهق:                    |
| 🎹  | :person_fencing:            | :شخص سياج:                        | 🎹   | :musical_keyboard:              | :لوحة المفاتيح الموسيقية:     |
| 😓  | :anguished_face:            | :الوجه المنعش:                    | 🍹   | :tropical_drink:                | :مشروب استوائي:               |
| 🔊  | :loudspeaker:               | :مكبر الصوت:                      | ❤   | :red_heart:                     | :قلب أحمر:                    |
| 🔇  | :muted_speaker:             | :مكبر صوت صامت:                   | ⬜   | :dotted_six-pointed_star:       | :نجم منقط من ستة نقاط:        |
| 🍏  | :green_apple:               | :تفاحة خضراء:                     | 😘   | :face_blowing_a_kiss:           | :الوجه ينفخ قبلة:             |
| 🎄  | :Christmas_tree:            | :شجرة عيد الميلاد:                | 🎄   | :Christmas_tree:                |                               |
| 🎵  | :musical_notes:             | :ملاحظات موسيقية:                 | 🙊   | :lying_face:                    | :الوجه الكذب:                 |
| 👩  | :woman_with_headscarf:      | :امرأة مع الحجاب:                 | 🗣   | :speaking_head:                 | :رئيس التحدث:                 |
| 🏃  | :dashing_away:              | :محطات بعيدا:                     | 👸   | :princess:                      | :أميرة:                       |
| 🍗  | :poultry_leg:               | :ساق الدواجن:                     | 👊   | :oncoming_fist:                 | :القبضة القادمة:              |
| 🛑  | :stop_sign:                 | :لافتة توقف:                      | ~   | ~                               | ~                             |
| 👠  | :woman's_sandal:            | :صندل المرأة:                     | -   | -                               | -                             |
| !? | :exclamation_question_mark: | :علامة استفهام تعجب:              | 💳   | :credit_card:                   | :بطاقة إئتمان:                |
| 👤  | :person_with_veil:          | :شخص مع حجاب:                     | 📄   | :page_facing_up:                | :صفحة مواجهة:                 |

|    |                                |                                   |    |                       |                         |
|----|--------------------------------|-----------------------------------|----|-----------------------|-------------------------|
| ☐  | :eagle:                        | :نسر:                             | ✋  | :OK_hand:             | :طيب اليد:              |
| <  | <                              | <                                 | ☐  | :nauseated_face:      | :الوجه الغثيان:         |
| P  | P                              | P                                 | ☐  | :cheese_wedge:        | :الجبن إسفين:           |
| ☐  | :glass_of_milk:                | :كوب من الحليب:                   | 🍺  | :beer_mug:            | :القدح البيرة:          |
| 🙈  | :hear-no-evil_monkey:          | :قرود لا يسمعون:                  | 🌺  | :hibiscus:            | :كرنديه:                |
| ”  | ”                              |                                   | 📞  | :telephone_receiver:  | :جهاز استقبال الهاتف:   |
| 🌙  | :crescent_moon:                | :الهلال:                          | 💫  | :sparkles:            | :التألق:                |
| 👂  | :ear:                          | :أذن:                             | ✈️ | :airplane:            | :طائرة:                 |
| 🐭  | :mouse_face:                   | :وجه الماوس:                      | ر  | ر                     | ر                       |
| 👁️ | :squinting_face_with_tongue:   | :تحلق الوجه مع اللسان:            | ★  | :star:                | :نجمة:                  |
| 🚊  | :railway_track:                | :مسار السكك الحديدية:             | 😬  | :grimacing_face:      | :الوجه المتجول:         |
| ♣️ | :shamrock:                     | :شامروك:                          | !  | !                     | !                       |
| 🎬  | :clapper_board:                | :لوحة كلاير:                      | 🎉  | :confetti_ball:       | :كرة الحلوى:            |
| 🍏  | :red_apple:                    | :تفاحة حمراء:                     | ☐  | :coconut:             | :جوزة الهند:            |
| 🚄  | :high-speed_train:             | :قطار شديد السرعة:                | ✍️ | :writing_hand:        | :كتابة اليد:            |
| \$ | \$                             | \$                                | 🐣  | :hatching_chick:      | :فرخ الفقس:             |
| 🚀  | :rocket:                       | :صاروخ:                           | 🐶  | :dog:                 | :كلب:                   |
| 💱  | :currency_exchange:            | :تحويل العملات:                   | 💬  | :speech_balloon:      | :بالون الكلام:          |
| ☐  | :drooling_face:                | :وجه سال لعابه:                   | 😐  | :expressionless_face: | :الوجه بدون تعبير:      |
| 📜  | :scroll:                       | :التمرير:                         | 🐵  | :monkey_face:         | :وجه القرد:             |
| %  | %                              | %                                 | ⚠️ | :warning:             | :تحذير:                 |
| 🕶️ | :smiling_face_with_sunglasses: | :الوجه المبتسم بالنظارات الشمسية: | 📢  | :megaphone:           | :مكبر الصوت:            |
| ①  | :ten-thirty:                   | :1030:                            | 😺  | :cat_with_wry_smile:  | :قطعة مع ابتسامة ساخرة: |
| ★  | :glowing_star:                 | :نجم متوهج:                       | 🎵  | :musical_score:       | :النتيجة الموسيقية:     |
| 🌯  | :burrito:                      | :بوريتو:                          | ❤️ | :black_heart:         | :قلب أسود:              |
| 💣  | :bomb:                         | :قنبلة:                           | 😬  | :flushed_face:        | :وجه متوهج:             |
| 👹  | :goblin:                       | :غفريت:                           | 🐷  | :pig:                 | :خنزير:                 |
| 😬  | :face_without_mouth:           | :وجه بدون فم:                     | 🚇  | :metro:               | :المترو:                |

|   |                                  |                                       |   |                                |                                  |
|---|----------------------------------|---------------------------------------|---|--------------------------------|----------------------------------|
| 🙄 | :unamused_face:                  | :وجه غير مستقر :                      | 🌐 | :globe_showing_Asia-Australia: | : تظهر آسيا :<br>:الأسترالية     |
| « | :«                               | :«                                    | 🎆 | :fireworks:                    | :العاب ناريه:                    |
| 🕊 | :dove:                           | :حمامة:                               | 🍆 | :eggplant:                     | :الباذنجان:                      |
| 🗑 | :wastebasket:                    | :سلة النفايات:                        | 🌀 | :spiral_shell:                 | :قذيفة حلزونية :                 |
| 💎 | :gem_stone:                      | :حجر الجوهرة :                        | 🤰 | :pregnant_woman:               | :امراة حامل :                    |
| 💥 | :collision:                      | :الاصطدام :                           | 🎓 | :graduation_cap:               | :قبعة التخرج:                    |
| 🧣 | :scarf:                          | :وشاح :                               | ؟ | ؟                              | ؟                                |
| 🎈 | :balloon:                        | :بالون :                              | 🌳 | :deciduous_tree:               | :شجرة متهاوية:                   |
| 👏 | :clapping_hands:                 | :التصفيق:                             | 💜 | :purple_heart:                 | :قلب بنفسجي:                     |
| 💧 | :sweat_droplets:                 | :قطرات العرق :                        | 🎯 | :bullseye:                     | :إصابة دقيقة للهدف:              |
| 💵 | :heavy_dollar_sign:              | :علامة الدولار الثقيل :               | 💻 | :laptop:                       | :حاسوب محمول:                    |
| 😓 | :anxious_face_with_sweat:        | :وجه قلق مع العرق :                   | 👉 | :backhand_index_pointing_down: | :فهرس الظهر يشير إلى :<br>:أسفل  |
| 🚿 | :shower:                         | :دش:                                  | 😺 | :smiling_cat_with_heart-eyes:  | :قطعة مبتسمة مع عيون :<br>:القلب |
| 🐷 | :pig_nose:                       | :أنف خنزير:                           | 🐵 | :monkey:                       | :فرد:                            |
| 🐎 | :horse:                          | :حصان:                                | 😬 | :face_with_tongue:             | :وجه مع اللسان :                 |
| 🌿 | :herb:                           | :عشب:                                 | 😷 | :face_with_medical_mask:       | :وجه مع قناع طبي :               |
| 👎 | :thumbs_down:                    | :استهجن:                              | ↓ | :down_arrow:                   | :السهم لأسفل:                    |
| 🍦 | :ice_cream:                      | :بوظة:                                | 👤 | :zipper-mouth_face:            | :الوجه الغم السحاب :             |
| ☢ | :radioactive:                    | :المشعة :                             | ⚡ |                                |                                  |
| 🌸 | :blossom:                        | : Blossom:                            | 👤 | :dark_skin_tone:               | :لون البشرة الداكن :             |
| 🛡 | :guard:                          | :يحمي:                                | 🍯 | :honey_pot:                    | :وعاء العسل:                     |
| ♂ | :male_sign:                      | :علامة الذكور :                       | 😺 | :pouting_cat:                  | :قطعة العبوس :                   |
| 🐩 | :poodle:                         | : Poodle:                             | 😺 | :kissing_cat:                  | :تقبيل القط :                    |
| 🔄 | :counterclockwise_arrows_button: | : زر سهم عكس اتجاه :<br>:عقارب الساعة | A | A                              | A                                |
| 🚬 | :cigarette:                      | :سيجارة:                              | 🤝 | :handshake:                    | :مصافحة:                         |
| 😘 | :kissing_face_with_smiling_eyes: | :تقبيل الوجه بعيون تبسم :             | ! | :red_exclamation_mark:         | :علامة التعجب الأحمر :           |
| 🐢 | :turtle:                         | :سلحفاة:                              | 🔨 | :hammer:                       | :شاكوش:                          |

|                                                                                     |                              |                                               |                                                                                     |                                                                                   |                                                                                     |
|-------------------------------------------------------------------------------------|------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|    | :confused_face:              | :الوجه المشوش :                               |    | :person_running:                                                                  | :شخص يركض :                                                                         |
|    | :down-left_arrow:            | :سهم أسفل اليسار :                            |    | :person_gesturing_OK:                                                             | :شخص يشير موافق :                                                                   |
|    | :tornado:                    | :إعصار:                                       |    | :astonished_face:                                                                 | :وجه مذهل :                                                                         |
|    | :fox:                        | :فوكس :                                       |    | :postal_horn:                                                                     | :القرن البريدي :                                                                    |
|    | :rabbit_face:                | :وجه الأرنب :                                 |    |  |  |
|    | :kitchen_knife:              | :سكين المطبخ:                                 |    | :duck:                                                                            | :بطة:                                                                               |
|                                                                                     |                              |                                               |    | :black_small_square:                                                              | :مربع صغير أسود :                                                                   |
|    | :clockwise_vertical_arrows:  | :المسهم الرأسية في اتجاه :<br>:عقارب الساعة : |    | :repeat_button:                                                                   | :زر التكرار :                                                                       |
|    | :camel:                      | :جمل:                                         |    | :medium-light_skin_tone:                                                          | :لون البشرة متوسطة :<br>:الضوء :                                                    |
|    | :face_with_crossed-out_eyes: | :وجه بعيون متقاطعة :                          |    | :kiwi_fruit:                                                                      | :فاكهة الكيوي:                                                                      |
|    | :woman:                      | :امرأة:                                       |    | :middle_finger:                                                                   | :الاصبع الوسطى:                                                                     |
|    | :ambulance:                  | :سياره اسعاف:                                 |    | :busts_in_silhouette:                                                             | :تمثال نصفي في صورة :<br>:ظلية :                                                    |
|    | :palm_tree:                  | :شجرة النخيل :                                |    | :tulip:                                                                           | :توليب :                                                                            |
|    | :cherry_blossom:             | :زهرة الكرز:                                  |    | :bouquet:                                                                         | :باقة أزهار:                                                                        |
|   | :baby_bottle:                | :زجاجة الطفل:                                 |   | :droplet:                                                                         | :قطيرة:                                                                             |
|  | :water_pistol:               | :ممسدس ماء:                                   |  | :memo:                                                                            | :مذكرة:                                                                             |
|  | :running_shoe:               | :حذاء الجري:                                  |  | :cookie:                                                                          | :بسكويت:                                                                            |
|  | :two_hearts:                 | :قلبين:                                       |  | :penguin:                                                                         | :البطريق:                                                                           |
|  | :tongue:                     | :لسان:                                        |  |                                                                                   |                                                                                     |
|  | :upside-down_face:           | :الوجه رأسا على عقب :                         |  |                                                                                   |                                                                                     |
|  | :light_skin_tone:            | :لون البشرة الفاتحة :                         |  |                                                                                   |                                                                                     |
|  | :worried_face:               | :وجه قلق :                                    |  | :alien:                                                                           | :كائن فضائي:                                                                        |
|  | :party_popper:               | : Party Popper:                               |  | :wrench:                                                                          | :مفتاح الربط:                                                                       |
|  | :station:                    | :محطة:                                        |  | :grinning_face:                                                                   | :الوجه الابتسامة :                                                                  |
|  | :dress:                      | :فستان:                                       |  | :sheaf_of_rice:                                                                   | :حلق الأرز :                                                                        |
|  | :yellow_heart:               | :القلب الأصفر :                               |  |                                                                                   |                                                                                     |
|  | :leaf_fluttering_in_wind:    | :ورقة ترفرف في الريح :                        |  | :shield:                                                                          | :درع:                                                                               |

|    |                                 |                                    |    |                                   |                                  |
|----|---------------------------------|------------------------------------|----|-----------------------------------|----------------------------------|
| 👉  | :backhand_index_pointing_right: | : مؤشر الظهر الذي يشير إلى اليمين: | 🚆  | :railway_car:                     | :سيارة السكك الحديدية:           |
| 🤓  | :nerd_face:                     | : الوجه الطالب الذي يذاكر كثيرا:   | 📷  | :camera_with_flash:               | :كاميرا مع فلاش:                 |
| 🚫  | :prohibited:                    | :محظور:                            | 🐼  | :panda:                           | :الباندا:                        |
| 🐾  | :paw_prints:                    | :بمخالبه:                          | 😓  | :pensive_face:                    | :الوجه البارز:                   |
| 🌹  | :rose:                          | :وردة:                             | 🌹  | :rose:                            | :وردة:                           |
| 🔔  | :bell_with_slash:               | :جرس مع مائل:                      | ☹️ | :frowning_face:                   | :وجه عيوس:                       |
| ♿  | :wheelchair_symbol:             | :رمز الكرسي المتحرك:               | ⛽  | :fuel_pump:                       | :مضخة وقود:                      |
| 🏠  | :medium_skin_tone:              | :لون البشرة المتوسطة:              | 💣  | :exploding_head:                  | :تفجر الرأس:                     |
| 💙  | :blue_heart:                    | :قلب أزرق:                         | 👠  | :woman's_boot:                    | :حذاء المرأة:                    |
| 👁️ | :face_with_monocle:             | :monocle وجه مع:                   | 👶  | :baby:                            | :طفل:                            |
| 💋  | :kiss_mark:                     | :علامة قبلة:                       | 🎻  | :violin:                          | :كمان:                           |
| 😭  | :crying_face:                   | :وجه البكاء:                       | 🕒  | :hourglass_not_done:              | :لم تنته Hourglass:              |
| 👃  | :nose:                          | :أنف:                              | 🏰  | :church:                          | :كنيسة:                          |
| 😬  | :confounded_face:               | :الوجه المربك:                     | 😬  | :face_with_open_mouth:            | :وجه مع الفم المفتوح:            |
| 🤝  | :men_holding_hands:             | :الرجال يمسكون يديه:               | 🤠  | :cowboy_hat_face:                 | :الوجه قبعة رعاة البقر:          |
| !! | :double_exclamation_mark:       | :علامة تعجب مزدوجة:                | 🦉  | :owl:                             | :بومة:                           |
| 👤  | :person_getting_massage:        | :شخص يحصل على التدليك:             | 😄  | :grinning_face_with_smiling_eyes: | :ابتسامة الابتسامة بعيون مبهتمة: |
| 🙏  | :folded_hands:                  | :الأيدي المطوية:                   | 🙏  | :folded_hands:                    | :الأيدي المطوية:                 |
| 🐎  | :horse_face:                    | :وجه الحصان:                       | 🐑  | :ewe:                             | :نعجة:                           |
| 🔔  | :star-struck:                   | :Star Lock:                        | 🔔  | :bell:                            | :جرس:                            |
| ⋯  | :...                            | :...                               | ⚽  | :soccer_ball:                     | :كرة القدم:                      |
| 😄  | :smiling_face_with_halo:        | :الوجه المبتسم مع الهالة:          | 🎮  | :video_game:                      | :لعبة فيديو:                     |
| 🐈  | :cat:                           | :قطعة:                             | 🐈  | :cat:                             | :قطعة:                           |
|    |                                 |                                    | 🚌  | :bus:                             | :حافلة:                          |

## Appendix B

Impact on changing batch size

| Train batch size | Batch Test size | Training Time | Test Time | Epoch | Validation Loss | Macro F1 | Macro Precision | Macro Recall | Accuracy |
|------------------|-----------------|---------------|-----------|-------|-----------------|----------|-----------------|--------------|----------|
| 64               | 128             | 669.692       | 49.151    | 1     | 0.316119        | 0.875550 | 0.882103        | 0.870298     | 0.886503 |
|                  |                 |               |           | 2     | 0.335552        | 0.876535 | 0.877757        | 0.875373     | 0.886065 |
|                  |                 |               |           | 3     | 0.350311        | 0.870120 | 0.872075        | 0.868315     | 0.880368 |
|                  |                 |               |           | 4     | 0.360054        | 0.871480 | 0.872170        | 0.870811     | 0.881245 |
|                  |                 |               |           | 5     | 0.370373        | 0.873509 | 0.873802        | 0.873220     | 0.882997 |
|                  |                 |               |           | 1     | 0.123188        | 0.960338 | 0.959194        | 0.961526     | 0.963190 |
|                  |                 |               |           | 2     | 0.125035        | 0.957658 | 0.960067        | 0.955418     | 0.960999 |
|                  |                 |               |           | 3     | 0.127812        | 0.954461 | 0.955428        | 0.953523     | 0.957932 |
|                  |                 |               |           | 4     | 0.127276        | 0.953536 | 0.954255        | 0.952834     | 0.957055 |
|                  |                 |               |           | 5     | 0.127441        | 0.953536 | 0.954255        | 0.952834     | 0.957055 |
|                  |                 |               |           | 1     | 0.127138        | 0.961233 | 0.960535        | 0.961946     | 0.964051 |
|                  |                 |               |           | 2     | 0.134982        | 0.958501 | 0.956724        | 0.960395     | 0.961420 |
|                  |                 |               |           | 3     | 0.136646        | 0.951536 | 0.948898        | 0.954452     | 0.954844 |
|                  |                 |               |           | 4     | 0.141996        | 0.946124 | 0.941970        | 0.951091     | 0.949584 |
|                  |                 |               |           | 5     | 0.131262        | 0.951678 | 0.951913        | 0.951445     | 0.955283 |
| 32               | 64              | 700.656       | 44.864    | 0     | 0.344713        | 0.854497 | 0.853789        | 0.855231     | 0.865031 |
|                  |                 |               |           | 2     | 0.365008        | 0.858201 | 0.858867        | 0.857556     | 0.868975 |
|                  |                 |               |           | 2     | 0.413130        | 0.846279 | 0.864191        | 0.835445     | 0.862840 |
|                  |                 |               |           | 4     | 0.425650        | 0.852183 | 0.850359        | 0.854197     | 0.862401 |
|                  |                 |               |           | 4     | 0.435287        | 0.856602 | 0.856511        | 0.856695     | 0.867222 |
|                  |                 |               |           | 0     | 0.154175        | 0.944455 | 0.945771        | 0.943195     | 0.948729 |
|                  |                 |               |           | 2     | 0.164015        | 0.936415 | 0.941193        | 0.932266     | 0.941718 |
|                  |                 |               |           | 2     | 0.168998        | 0.935029 | 0.935836        | 0.934244     | 0.939965 |
|                  |                 |               |           | 4     | 0.170792        | 0.934318 | 0.933555        | 0.935103     | 0.939089 |
|                  |                 |               |           | 4     | 0.169628        | 0.933448 | 0.935350        | 0.931662     | 0.938650 |
|                  |                 |               |           | 0     | 0.155453        | 0.945017 | 0.945483        | 0.944558     | 0.949145 |
|                  |                 |               |           | 1     | 0.175874        | 0.936421 | 0.931701        | 0.942307     | 0.940377 |
|                  |                 |               |           | 2     | 0.169640        | 0.933267 | 0.930296        | 0.936630     | 0.937747 |

|    |    |         |        |   |          |          |          |          |          |
|----|----|---------|--------|---|----------|----------|----------|----------|----------|
|    |    |         |        | 3 | 0.167478 | 0.934200 | 0.934089 | 0.934312 | 0.939062 |
|    |    |         |        | 4 | 0.169438 | 0.930159 | 0.931674 | 0.928720 | 0.935555 |
| 16 | 32 | 814.509 | 53.506 | 1 | 0.516701 | 0.735896 | 0.759937 | 0.726228 | 0.769500 |
|    |    |         |        | 2 | 0.381495 | 0.819185 | 0.830215 | 0.811774 | 0.836985 |
|    |    |         |        | 3 | 0.368503 | 0.838505 | 0.844574 | 0.833718 | 0.852761 |
|    |    |         |        | 4 | 0.387293 | 0.841915 | 0.847910 | 0.837161 | 0.855828 |
|    |    |         |        | 5 | 0.401875 | 0.836903 | 0.837438 | 0.836383 | 0.849255 |
|    |    |         |        | 1 | 0.175797 | 0.940843 | 0.940729 | 0.940956 | 0.945223 |
|    |    |         |        | 2 | 0.175962 | 0.930258 | 0.934541 | 0.926499 | 0.936021 |
|    |    |         |        | 3 | 0.183914 | 0.923281 | 0.926281 | 0.920560 | 0.929448 |
|    |    |         |        | 4 | 0.194920 | 0.920379 | 0.923621 | 0.917462 | 0.926819 |
|    |    |         |        | 5 | 0.203451 | 0.922405 | 0.922189 | 0.922624 | 0.928133 |
|    |    |         |        | 1 | 0.170749 | 0.942780 | 0.942220 | 0.943351 | 0.946953 |
|    |    |         |        | 2 | 0.176791 | 0.931740 | 0.929403 | 0.934307 | 0.936431 |
|    |    |         |        | 3 | 0.173517 | 0.928221 | 0.929974 | 0.926568 | 0.933801 |
|    |    |         |        | 4 | 0.181366 | 0.926096 | 0.926317 | 0.925876 | 0.931609 |
|    |    |         |        | 5 | 0.195457 | 0.922879 | 0.922556 | 0.923206 | 0.928540 |

## Appendix C

Impact on changing LSTM hidden size and layers

| Hidden size | layer | Epoch | Validation Loss | Macro F1 | Accuracy |
|-------------|-------|-------|-----------------|----------|----------|
| 32          | 1     | 1     | 0.351782        | 0.849644 | 0.864134 |
|             |       | 2     | 0.370725        | 0.847168 | 0.861213 |
|             |       | 3     | 0.37788         | 0.850212 | 0.861213 |
|             |       | 4     | 0.382388        | 0.845718 | 0.859021 |
|             |       | 1     | 0.173114        | 0.938897 | 0.943755 |
|             |       | 2     | 0.200045        | 0.923439 | 0.928415 |
|             |       | 3     | 0.182494        | 0.936005 | 0.940833 |
|             |       | 4     | 0.190732        | 0.933778 | 0.938641 |
|             |       | 1     | 0.09264         | 0.971643 | 0.973703 |
|             |       | 2     | 0.095705        | 0.964258 | 0.967129 |
|             |       | 3     | 0.091222        | 0.972226 | 0.974434 |
|             |       | 4     | 0.097628        | 0.967663 | 0.970051 |
| 32          | 6     | 0     | 0.55135         | 0.65951  | 0.72568  |
|             |       | 1     | 0.40612         | 0.7972   | 0.81683  |
|             |       | 2     | 0.38479         | 0.81387  | 0.83567  |
|             |       | 3     | 0.36994         | 0.84008  | 0.85188  |
|             |       | 4     | 0.37523         | 0.84114  | 0.85364  |
|             |       | 0     | 0.21569         | 0.91552  | 0.922    |
|             |       | 1     | 0.2169          | 0.91045  | 0.91805  |
|             |       | 2     | 0.2216          | 0.9029   | 0.91104  |
|             |       | 3     | 0.22395         | 0.90743  | 0.91455  |
|             |       | 4     | 0.22554         | 0.90638  | 0.91367  |
|             |       | 0     | 0.2073          | 0.92219  | 0.9281   |
|             |       | 1     | 0.21642         | 0.91798  | 0.92328  |
|             |       | 2     | 0.21369         | 0.91223  | 0.91802  |
|             |       | 3     | 0.21326         | 0.91291  | 0.9189   |
|             |       | 4     | 0.21283         | 0.90959  | 0.91627  |
| 32          | 4     | 0     | 0.5412          | 0.6899   | 0.74628  |
|             |       | 1     | 0.39992         | 0.81074  | 0.82734  |
|             |       | 2     | 0.37978         | 0.81932  | 0.84005  |

|     |   |   |          |          |          |
|-----|---|---|----------|----------|----------|
|     |   | 3 | 0.36611  | 0.83888  | 0.85101  |
|     |   | 4 | 0.37265  | 0.84167  | 0.85451  |
|     |   | 0 | 0.20807  | 0.92241  | 0.92857  |
|     |   | 1 | 0.21041  | 0.91483  | 0.922    |
|     |   | 2 | 0.21428  | 0.91149  | 0.91849  |
|     |   | 3 | 0.21993  | 0.90934  | 0.91586  |
|     |   | 4 | 0.21831  | 0.90892  | 0.91586  |
|     |   | 0 | 0.20713  | 0.91942  | 0.92547  |
|     |   | 1 | 0.21831  | 0.91248  | 0.91802  |
|     |   | 2 | 0.21356  | 0.9091   | 0.91539  |
|     |   | 3 | 0.21395  | 0.9085   | 0.91539  |
|     |   | 4 | 0.21607  | 0.90944  | 0.91627  |
| 32  | 2 | 0 | 0.34116  | 0.83585  | 0.85238  |
|     |   | 1 | 0.34314  | 0.84406  | 0.85634  |
|     |   | 0 | 0.30477  | 0.85856  | 0.87161  |
|     |   | 1 | 0.30537  | 0.86048  | 0.8733   |
|     |   | 0 | 0.30477  | 0.85856  | 0.87161  |
|     |   | 1 | 0.30537  | 0.86048  | 0.8733   |
|     |   | 0 | 0.266738 | 0.883972 | 0.894796 |
|     |   | 1 | 0.264065 | 0.88594  | 0.897059 |
|     |   | 0 | 0.23259  | 0.90606  | 0.91572  |
|     |   | 1 | 0.23109  | 0.90371  | 0.9129   |
| 128 | 2 | 0 | 0.52147  | 0.72339  | 0.76249  |
|     |   | 1 | 0.38637  | 0.81808  | 0.83129  |
|     |   | 2 | 0.3712   | 0.83447  | 0.85013  |
|     |   | 3 | 0.37172  | 0.83595  | 0.84926  |
|     |   | 0 | 0.2458   | 0.90125  | 0.90929  |
|     |   | 1 | 0.24669  | 0.89489  | 0.90403  |
|     |   | 2 | 0.25033  | 0.89189  | 0.9014   |
|     |   | 3 | 0.2534   | 0.8943   | 0.90272  |
|     |   | 0 | 0.23675  | 0.90904  | 0.91583  |
|     |   | 1 | 0.24637  | 0.90333  | 0.90969  |
|     |   | 2 | 0.24639  | 0.90324  | 0.90969  |
|     |   | 3 | 0.24927  | 0.89973  | 0.90662  |

## Appendix D

This table shows the top 500 words without emoji and the percentage of repeating each word mentioned in OSACT dataset

| WORD    | COUNT | PERCENTAGE   |
|---------|-------|--------------|
| مستخدم  | 4439  | 4.71957897   |
| –       | 3097  | 3.29275424   |
| من      | 1402  | 1.490617192  |
| URL     | 1213  | 1.289670937  |
| NOT     | 1175  | 1.249269045  |
| RT      | 913   | 0.9707086279 |
| الله    | 815   | 0.8665142736 |
| في      | 712   | 0.7570038807 |
| الكليجا | 703   | 0.7474350114 |
| بايع    | 651   | 0.6921482112 |
| اللي    | 567   | 0.6028387646 |
| و       | 562   | 0.5975227261 |
| على     | 560   | 0.5953963107 |
| ما      | 530   | 0.5635000797 |
| بس      | 520   | 0.5528680028 |
| لا      | 496   | 0.527351018  |
| انا     | 450   | 0.4784434639 |
| والله   | 413   | 0.4391047791 |
| ولا     | 408   | 0.4337887406 |
| يا      | 405   | 0.4305991175 |
| OFF     | 330   | 0.3508585402 |
| كل      | 330   | 0.3508585402 |
| VLG     | 330   | 0.3508585402 |
| VIO     | 330   | 0.3508585402 |
| HS      | 294   | 0.3125830631 |

|        |     |              |
|--------|-----|--------------|
| انت    | 277 | 0.2945085322 |
| ع      | 274 | 0.2913189091 |
| الي    | 257 | 0.2732443783 |
| لو     | 251 | 0.2668651321 |
| فيه    | 243 | 0.2583594705 |
| مع     | 240 | 0.2551698474 |
| مو     | 240 | 0.2551698474 |
| هذا    | 235 | 0.2498538089 |
| وش     | 225 | 0.239221732  |
| شي     | 224 | 0.2381585243 |
| ان     | 209 | 0.2222104088 |
| هه     | 208 | 0.2211472011 |
| عن     | 207 | 0.2200839934 |
| البنات | 185 | 0.1966934241 |
| اذا    | 183 | 0.1945670087 |
| مش     | 181 | 0.1924405933 |
| هو     | 176 | 0.1871245548 |
| ي      | 175 | 0.1860613471 |
| بعد    | 174 | 0.1849981394 |
| عشان   | 169 | 0.1796821009 |
| حتى    | 165 | 0.1754292701 |
| الا    | 162 | 0.172239647  |
| م      | 157 | 0.1669236085 |
| يوم    | 156 | 0.1658604008 |
| الناس  | 155 | 0.1647971931 |
| يعني   | 154 | 0.1637339854 |
| اي     | 145 | 0.1541651162 |
| كان    | 143 | 0.1520387008 |
| عليه   | 138 | 0.1467226623 |
| او     | 130 | 0.1382170007 |
| بالله  | 127 | 0.1350273776 |
| علي    | 125 | 0.1329009622 |
| الرجال | 125 | 0.1329009622 |

|       |     |               |
|-------|-----|---------------|
| ف     | 123 | 0.1307745468  |
| ذا    | 122 | 0.1297113391  |
| واحد  | 120 | 0.1275849237  |
| كذا   | 120 | 0.1275849237  |
| بعض   | 117 | 0.1243953006  |
| انتي  | 116 | 0.1233320929  |
| لي    | 115 | 0.1222688852  |
| خلاص  | 114 | 0.1212056775  |
| لما   | 114 | 0.1212056775  |
| عليك  | 110 | 0.1169528467  |
| بنات  | 108 | 0.1148264313  |
| غير   | 106 | 0.1127000159  |
| وانا  | 105 | 0.1116368083  |
| ايه   | 101 | 0.1073839775  |
| له    | 98  | 0.1041943544  |
| ليش   | 96  | 0.102067939   |
| وين   | 95  | 0.1010047313  |
| فيها  | 94  | 0.09994152358 |
| بنت   | 94  | 0.09994152358 |
| لك    | 93  | 0.09887831588 |
| انه   | 91  | 0.09675190048 |
| كيف   | 90  | 0.09568869279 |
| اني   | 88  | 0.09356227739 |
| قال   | 86  | 0.091435862   |
| عيب   | 86  | 0.091435862   |
| قبل   | 85  | 0.0903726543  |
| مره   | 85  | 0.0903726543  |
| عاد   | 84  | 0.0893094466  |
| ليه   | 83  | 0.0882462389  |
| اليوم | 82  | 0.08718303121 |
| خير   | 82  | 0.08718303121 |
| كنت   | 82  | 0.08718303121 |
| هي    | 81  | 0.08611982351 |

|         |    |               |
|---------|----|---------------|
| ايش     | 79 | 0.08399340811 |
| عادي    | 76 | 0.08080378502 |
| لكن     | 75 | 0.07974057732 |
| مثل     | 73 | 0.07761416193 |
| ب       | 73 | 0.07761416193 |
| هنا     | 73 | 0.07761416193 |
| صباح    | 72 | 0.07655095423 |
| طيب     | 72 | 0.07655095423 |
| احد     | 72 | 0.07655095423 |
| علينا   | 71 | 0.07548774653 |
| صار     | 71 | 0.07548774653 |
| راح     | 69 | 0.07336133114 |
| ذي      | 68 | 0.07229812344 |
| منه     | 68 | 0.07229812344 |
| الحين   | 68 | 0.07229812344 |
| زي      | 67 | 0.07123491574 |
| ال      | 66 | 0.07017170804 |
| وحده    | 66 | 0.07017170804 |
| عليكم   | 66 | 0.07017170804 |
| هذي     | 65 | 0.06910850035 |
| ناس     | 65 | 0.06910850035 |
| الحياء  | 65 | 0.06910850035 |
| انك     | 65 | 0.06910850035 |
| ده      | 65 | 0.06910850035 |
| القادمه | 64 | 0.06804529265 |
| لها     | 64 | 0.06804529265 |
| لازم    | 64 | 0.06804529265 |
| الى     | 63 | 0.06698208495 |
| يقول    | 63 | 0.06698208495 |
| عند     | 63 | 0.06698208495 |
| الجنازه | 63 | 0.06698208495 |
| أن      | 62 | 0.06591887725 |
| صلاه    | 62 | 0.06591887725 |

|       |    |               |
|-------|----|---------------|
| حد    | 62 | 0.06591887725 |
| جدا   | 59 | 0.06272925416 |
| رجال  | 59 | 0.06272925416 |
| هم    | 59 | 0.06272925416 |
| حلو   | 59 | 0.06272925416 |
| شيء   | 59 | 0.06272925416 |
| شوي   | 58 | 0.06166604646 |
| اول   | 58 | 0.06166604646 |
| يارب  | 58 | 0.06166604646 |
| نفس   | 58 | 0.06166604646 |
| ام    | 57 | 0.06060283876 |
| عليها | 57 | 0.06060283876 |
| وربي  | 57 | 0.06060283876 |
| أنا   | 56 | 0.05953963107 |
| عندي  | 56 | 0.05953963107 |
| ترا   | 56 | 0.05953963107 |
| منهم  | 56 | 0.05953963107 |
| وهو   | 55 | 0.05847642337 |
| قلة   | 54 | 0.05741321567 |
| نفسك  | 53 | 0.05635000797 |
| يع    | 52 | 0.05528680028 |
| تقول  | 52 | 0.05528680028 |
| اكثر  | 52 | 0.05528680028 |
| سنه   | 51 | 0.05422359258 |
| حياء  | 50 | 0.05316038488 |
| بعدين | 50 | 0.05316038488 |
| مدري  | 50 | 0.05316038488 |
| حسبي  | 50 | 0.05316038488 |
| قسم   | 49 | 0.05209717718 |
| علشان | 49 | 0.05209717718 |
| كده   | 49 | 0.05209717718 |
| بدون  | 48 | 0.05103396949 |
| ابو   | 48 | 0.05103396949 |

|        |    |               |
|--------|----|---------------|
| جد     | 48 | 0.05103396949 |
| إذا    | 48 | 0.05103396949 |
| وانت   | 48 | 0.05103396949 |
| العالم | 48 | 0.05103396949 |
| دا     | 47 | 0.04997076179 |
| شفت    | 47 | 0.04997076179 |
| بكل    | 47 | 0.04997076179 |
| اقول   | 45 | 0.04784434639 |
| كله    | 45 | 0.04784434639 |
| اصلا   | 44 | 0.0467811387  |
| صدق    | 44 | 0.0467811387  |
| انها   | 44 | 0.0467811387  |
| البنت  | 44 | 0.0467811387  |
| ومن    | 44 | 0.0467811387  |
| زين    | 43 | 0.045717931   |
| عليهم  | 43 | 0.045717931   |
| احنا   | 43 | 0.045717931   |
| إلا    | 42 | 0.0446547233  |
| قاعد   | 42 | 0.0446547233  |
| دي     | 42 | 0.0446547233  |
| انتم   | 42 | 0.0446547233  |
| الشباب | 42 | 0.0446547233  |
| نفسي   | 42 | 0.0446547233  |
| كانت   | 41 | 0.0435915156  |
| دخل    | 41 | 0.0435915156  |
| لم     | 41 | 0.0435915156  |
| شخص    | 41 | 0.0435915156  |
| متى    | 41 | 0.0435915156  |
| حق     | 41 | 0.0435915156  |
| نفسه   | 41 | 0.0435915156  |
| تكون   | 40 | 0.0425283079  |
| يكون   | 40 | 0.0425283079  |
| لهم    | 40 | 0.0425283079  |

|         |    |               |
|---------|----|---------------|
| الموضوع | 40 | 0.0425283079  |
| اخر     | 40 | 0.0425283079  |
| كلب     | 40 | 0.0425283079  |
| صح      | 40 | 0.0425283079  |
| عندك    | 40 | 0.0425283079  |
| أو      | 40 | 0.0425283079  |
| كلام    | 40 | 0.0425283079  |
| ل       | 40 | 0.0425283079  |
| فيهم    | 39 | 0.04146510021 |
| بين     | 39 | 0.04146510021 |
| يصير    | 39 | 0.04146510021 |
| ياالله  | 39 | 0.04146510021 |
| اشوف    | 39 | 0.04146510021 |
| بريد    | 38 | 0.04040189251 |
| معنفك   | 38 | 0.04040189251 |
| حياتك   | 38 | 0.04040189251 |
| وهي     | 38 | 0.04040189251 |
| تويتر   | 38 | 0.04040189251 |
| فيك     | 38 | 0.04040189251 |
| محد     | 38 | 0.04040189251 |
| مافي    | 38 | 0.04040189251 |
| كلهم    | 37 | 0.03933868481 |
| اما     | 37 | 0.03933868481 |
| ادب     | 37 | 0.03933868481 |
| واضح    | 36 | 0.03827547711 |
| عندنا   | 36 | 0.03827547711 |
| المفروض | 36 | 0.03827547711 |
| احسن    | 36 | 0.03827547711 |
| لين     | 36 | 0.03827547711 |
| إن      | 35 | 0.03721226942 |
| تطلع    | 35 | 0.03721226942 |
| شو      | 35 | 0.03721226942 |
| قلبي    | 35 | 0.03721226942 |

|        |    |               |
|--------|----|---------------|
| ترامب  | 35 | 0.03721226942 |
| ولد    | 35 | 0.03721226942 |
| يجي    | 35 | 0.03721226942 |
| فعلا   | 35 | 0.03721226942 |
| افضح   | 35 | 0.03721226942 |
| مكان   | 34 | 0.03614906172 |
| عنده   | 34 | 0.03614906172 |
| الكلام | 34 | 0.03614906172 |
| ربي    | 34 | 0.03614906172 |
| قلت    | 34 | 0.03614906172 |
| هل     | 34 | 0.03614906172 |
| مب     | 34 | 0.03614906172 |
| واللي  | 34 | 0.03614906172 |
| حيا    | 34 | 0.03614906172 |
| قروب   | 33 | 0.03508585402 |
| انهم   | 33 | 0.03508585402 |
| قالت   | 33 | 0.03508585402 |
| اللى   | 33 | 0.03508585402 |
| حاجه   | 33 | 0.03508585402 |
| مين    | 33 | 0.03508585402 |
| الدنيا | 33 | 0.03508585402 |
| ماعاد  | 32 | 0.03402264632 |
| وجه    | 32 | 0.03402264632 |
| 3      | 32 | 0.03402264632 |
| تحت    | 32 | 0.03402264632 |
| شنو    | 32 | 0.03402264632 |
| الكويت | 32 | 0.03402264632 |
| يع     | 32 | 0.03402264632 |
| غلط    | 32 | 0.03402264632 |
| ياخي   | 32 | 0.03402264632 |
| وبعدين | 32 | 0.03402264632 |
| بلا    | 32 | 0.03402264632 |
| كم     | 32 | 0.03402264632 |

|           |    |               |
|-----------|----|---------------|
| فى        | 32 | 0.03402264632 |
| سالب      | 32 | 0.03402264632 |
| شاء       | 31 | 0.03295943863 |
| لكم       | 31 | 0.03295943863 |
| ابي       | 31 | 0.03295943863 |
| خاص       | 31 | 0.03295943863 |
| الخير     | 31 | 0.03295943863 |
| لنا       | 30 | 0.03189623093 |
| وقت       | 30 | 0.03189623093 |
| حرام      | 30 | 0.03189623093 |
| اجل       | 30 | 0.03189623093 |
| ترى       | 30 | 0.03189623093 |
| مالي      | 30 | 0.03189623093 |
| الاهلي    | 30 | 0.03189623093 |
| لان       | 30 | 0.03189623093 |
| الحمد لله | 30 | 0.03189623093 |
| فيكم      | 29 | 0.03083302323 |
| فوق       | 29 | 0.03083302323 |
| كدا       | 29 | 0.03083302323 |
| يبي       | 29 | 0.03083302323 |
| طبيعي     | 29 | 0.03083302323 |
| الان      | 29 | 0.03083302323 |
| ربنا      | 29 | 0.03083302323 |
| رتويت     | 29 | 0.03083302323 |
| هذه       | 29 | 0.03083302323 |
| الواحد    | 29 | 0.03083302323 |
| وهذا      | 28 | 0.02976981553 |
| نعم       | 28 | 0.02976981553 |
| للاسف     | 28 | 0.02976981553 |
| طول       | 28 | 0.02976981553 |
| منها      | 28 | 0.02976981553 |
| الحب      | 28 | 0.02976981553 |
| بانع      | 28 | 0.02976981553 |

|         |    |               |
|---------|----|---------------|
| المشكلة | 28 | 0.02976981553 |
| به      | 28 | 0.02976981553 |
| عنه     | 28 | 0.02976981553 |
| عم      | 28 | 0.02976981553 |
| فقط     | 27 | 0.02870660784 |
| قد      | 27 | 0.02870660784 |
| 8       | 27 | 0.02870660784 |
| اربع    | 27 | 0.02870660784 |
| والا    | 27 | 0.02870660784 |
| للأسف   | 27 | 0.02870660784 |
| ممکن    | 27 | 0.02870660784 |
| الذي    | 27 | 0.02870660784 |
| طب      | 27 | 0.02870660784 |
| بعيد    | 27 | 0.02870660784 |
| وبينهم  | 27 | 0.02870660784 |
| الوضع   | 27 | 0.02870660784 |
| حب      | 27 | 0.02870660784 |
| الكلب   | 27 | 0.02870660784 |
| للدعم   | 27 | 0.02870660784 |
| حياتي   | 26 | 0.02764340014 |
| 1       | 26 | 0.02764340014 |
| وجهك    | 26 | 0.02764340014 |
| 2       | 26 | 0.02764340014 |
| وجهه    | 26 | 0.02764340014 |
| كلها    | 26 | 0.02764340014 |
| عندهم   | 26 | 0.02764340014 |
| وكل     | 26 | 0.02764340014 |
| قطر     | 26 | 0.02764340014 |
| انو     | 26 | 0.02764340014 |
| هـ      | 26 | 0.02764340014 |
| صارت    | 25 | 0.02658019244 |
| عييب    | 25 | 0.02658019244 |
| وقلة    | 25 | 0.02658019244 |

|         |    |               |
|---------|----|---------------|
| 23      | 25 | 0.02658019244 |
| الجو    | 25 | 0.02658019244 |
| أنت     | 25 | 0.02658019244 |
| اكيد    | 25 | 0.02658019244 |
| اوفر    | 25 | 0.02658019244 |
| شكله    | 25 | 0.02658019244 |
| منك     | 25 | 0.02658019244 |
| الرجل   | 25 | 0.02658019244 |
| ا       | 25 | 0.02658019244 |
| امس     | 25 | 0.02658019244 |
| يمكن    | 25 | 0.02658019244 |
| اللهم   | 25 | 0.02658019244 |
| صور     | 25 | 0.02658019244 |
| معه     | 25 | 0.02658019244 |
| الرياض  | 25 | 0.02658019244 |
| بسبب    | 25 | 0.02658019244 |
| الصبح   | 25 | 0.02658019244 |
| اكره    | 24 | 0.02551698474 |
| ودي     | 24 | 0.02551698474 |
| الوناسه | 24 | 0.02551698474 |
| طلع     | 24 | 0.02551698474 |
| يقولون  | 24 | 0.02551698474 |
| يطلع    | 24 | 0.02551698474 |
| العيال  | 24 | 0.02551698474 |
| اريد    | 24 | 0.02551698474 |
| وما     | 24 | 0.02551698474 |
| كليجا   | 24 | 0.02551698474 |
| اهم     | 24 | 0.02551698474 |
| ثاني    | 24 | 0.02551698474 |
| اعوذ    | 23 | 0.02445377705 |
| كمان    | 23 | 0.02445377705 |
| فك      | 23 | 0.02445377705 |
| مهجوره  | 23 | 0.02445377705 |

|         |    |               |
|---------|----|---------------|
| فييني   | 23 | 0.02445377705 |
| ترند    | 23 | 0.02445377705 |
| الف     | 23 | 0.02445377705 |
| اصدقا   | 23 | 0.02445377705 |
| النوم   | 23 | 0.02445377705 |
| وانتم   | 23 | 0.02445377705 |
| محمد    | 23 | 0.02445377705 |
| بشكل    | 23 | 0.02445377705 |
| لحد     | 23 | 0.02445377705 |
| فشلتونا | 23 | 0.02445377705 |
| يابنات  | 23 | 0.02445377705 |
| إلى     | 23 | 0.02445377705 |
| اكبر    | 23 | 0.02445377705 |
| طبعا    | 23 | 0.02445377705 |
| اه      | 23 | 0.02445377705 |
| جديد    | 23 | 0.02445377705 |
| جميل    | 23 | 0.02445377705 |
| كثر     | 23 | 0.02445377705 |
| مبارك   | 23 | 0.02445377705 |
| داليا   | 23 | 0.02445377705 |
| شعب     | 23 | 0.02445377705 |
| يلا     | 22 | 0.02339056935 |
| صحيح    | 22 | 0.02339056935 |
| الارض   | 22 | 0.02339056935 |
| سبحان   | 22 | 0.02339056935 |
| راعي    | 22 | 0.02339056935 |
| صراحه   | 22 | 0.02339056935 |
| مافيه   | 22 | 0.02339056935 |
| وجع     | 22 | 0.02339056935 |
| لمن     | 22 | 0.02339056935 |
| روح     | 22 | 0.02339056935 |
| استغفر  | 22 | 0.02339056935 |
| وع      | 22 | 0.02339056935 |

|          |    |               |
|----------|----|---------------|
| حاجة     | 22 | 0.02339056935 |
| ونعم     | 22 | 0.02339056935 |
| يلعن     | 22 | 0.02339056935 |
| جمال     | 22 | 0.02339056935 |
| شكرا     | 22 | 0.02339056935 |
| سنة      | 22 | 0.02339056935 |
| ضد       | 22 | 0.02339056935 |
| الدين    | 22 | 0.02339056935 |
| مصر      | 22 | 0.02339056935 |
| 10       | 21 | 0.02232736165 |
| بلوك     | 21 | 0.02232736165 |
| شكلك     | 21 | 0.02232736165 |
| غيره     | 21 | 0.02232736165 |
| قليلات   | 21 | 0.02232736165 |
| حلوه     | 21 | 0.02232736165 |
| كامل     | 21 | 0.02232736165 |
| رجل      | 21 | 0.02232736165 |
| زمان     | 21 | 0.02232736165 |
| الحياة   | 21 | 0.02232736165 |
| السعوديه | 21 | 0.02232736165 |
| ياربي    | 21 | 0.02232736165 |
| قيودي    | 21 | 0.02232736165 |
| كلمة     | 21 | 0.02232736165 |
| ماله     | 21 | 0.02232736165 |
| تعرف     | 21 | 0.02232736165 |
| الكل     | 21 | 0.02232736165 |
| باقي     | 21 | 0.02232736165 |
| زق       | 21 | 0.02232736165 |
| وفي      | 21 | 0.02232736165 |
| مرة      | 21 | 0.02232736165 |
| الوكيل   | 20 | 0.02126415395 |
| اقسم     | 20 | 0.02126415395 |
| يستر     | 20 | 0.02126415395 |

|          |    |               |
|----------|----|---------------|
| القرف    | 20 | 0.02126415395 |
| تم       | 20 | 0.02126415395 |
| منكم     | 20 | 0.02126415395 |
| النساء   | 20 | 0.02126415395 |
| الباطن   | 20 | 0.02126415395 |
| ماهو     | 20 | 0.02126415395 |
| ثانيه    | 20 | 0.02126415395 |
| مفيش     | 20 | 0.02126415395 |
| التاق    | 20 | 0.02126415395 |
| ثم       | 20 | 0.02126415395 |
| يفشلكم   | 20 | 0.02126415395 |
| مالك     | 20 | 0.02126415395 |
| كذب      | 20 | 0.02126415395 |
| اطلع     | 20 | 0.02126415395 |
| ديوث     | 20 | 0.02126415395 |
| اعرف     | 20 | 0.02126415395 |
| وحدة     | 20 | 0.02126415395 |
| عارف     | 20 | 0.02126415395 |
| سبب      | 20 | 0.02126415395 |
| السعودية | 19 | 0.02020094625 |
| عمه      | 19 | 0.02020094625 |
| كثير     | 19 | 0.02020094625 |
| وهم      | 19 | 0.02020094625 |
| لكل      | 19 | 0.02020094625 |
| مادري    | 19 | 0.02020094625 |
| احب      | 19 | 0.02020094625 |
| الدوري   | 19 | 0.02020094625 |
| لن       | 19 | 0.02020094625 |
| انتو     | 19 | 0.02020094625 |
| معي      | 19 | 0.02020094625 |
| HS6      | 19 | 0.02020094625 |
| تبي      | 19 | 0.02020094625 |
| الحيا    | 19 | 0.02020094625 |

|         |    |               |
|---------|----|---------------|
| تف      | 19 | 0.02020094625 |
| صورة    | 19 | 0.02020094625 |
| كمية    | 19 | 0.02020094625 |
| وعلى    | 19 | 0.02020094625 |
| ولكن    | 19 | 0.02020094625 |
| كبير    | 19 | 0.02020094625 |
| اسم     | 19 | 0.02020094625 |
| انكم    | 19 | 0.02020094625 |
| الولد   | 19 | 0.02020094625 |
| ماشاء   | 19 | 0.02020094625 |
| عليا    | 19 | 0.02020094625 |
| قدام    | 19 | 0.02020094625 |
| وأنا    | 19 | 0.02020094625 |
| حبيبي   | 19 | 0.02020094625 |
| 4       | 19 | 0.02020094625 |
| يجيب    | 19 | 0.02020094625 |
| تجي     | 19 | 0.02020094625 |
| الشعب   | 18 | 0.01913773856 |
| التي    | 18 | 0.01913773856 |
| مني     | 18 | 0.01913773856 |
| تعليق   | 18 | 0.01913773856 |
| عندكم   | 18 | 0.01913773856 |
| مسوي    | 18 | 0.01913773856 |
| الخاص   | 18 | 0.01913773856 |
| اعمل    | 18 | 0.01913773856 |
| المرأة  | 18 | 0.01913773856 |
| كلنا    | 18 | 0.01913773856 |
| غبار    | 18 | 0.01913773856 |
| مجتمع   | 18 | 0.01913773856 |
| حل      | 18 | 0.01913773856 |
| نص      | 18 | 0.01913773856 |
| الثانيه | 18 | 0.01913773856 |
| افضل    | 18 | 0.01913773856 |

|       |    |               |
|-------|----|---------------|
| بسم   | 18 | 0.01913773856 |
| اتمنى | 18 | 0.01913773856 |
| أنه   | 18 | 0.01913773856 |
| قرف   | 18 | 0.01913773856 |
| عقل   | 18 | 0.01913773856 |
| طلعت  | 18 | 0.01913773856 |
| استحي | 18 | 0.01913773856 |
| هناك  | 18 | 0.01913773856 |
| بكره  | 18 | 0.01913773856 |
| الاول | 17 | 0.01807453086 |
| عني   | 17 | 0.01807453086 |
| خلق   | 17 | 0.01807453086 |