Arabic Offensive Text Detection Using Machine Learning Algorithms and Cloud Computing

Material: Cloud Computing & Big Data
Team: Rami Ghanem, Musab Hamdan, Yazan Manasir
Supervision: Dr Heider Wahsheh
Semester: Second 2022/2023

Introduction

The detection of offensive text in Arabic language is a crucial task in Natural Language Processing (NLP). This project aims to develop an Arabic Offensive or Normal Text Classifier using Machine Learning Algorithms and cloud computing. By training machine learning models on a dataset of Arabic social media text, we will classify text as "normal" or "offensive" using algorithms such as Support Vector Machines (SVM) and Extreme Gradient Boosting (XGBoost).

With the increasing use of Arabic in various applications and services, there is a growing need for effective methods to identify and filter abusive language. This project addresses that need by leveraging cloud computing and big data technologies to improve the efficiency and accuracy of offensive text detection. Cloud storage and processing services, such as Google Drive and Colab, will be utilized to handle the vast amount of data available on the internet.

Detecting offensive text plays a crucial role in maintaining a safe and respectful online environment, particularly in the context of user-generated content on social media platforms. By developing an automated system to filter and identify potentially offensive language, we aim to contribute to the development of safer online environments and advance Arabic NLP research.

In summary, this project addresses the need for effective offensive text detection in Arabic language using machine learning algorithms and cloud computing. Through our efforts, we aspire to contribute to the development of safer online environments and the advancement of Arabic NLP research.

Background and Literature:

Arabic NLP is gaining a lot of attention by researchers because the Arabic language is being widely used in different applications and services and is one of the top five most used languages on the internet (Albirini, 2016). Therefore, efforts are being taken to increase Arabic content, Arabic-based search tools and other models and applications on the Internet.

The term "offensive text detection" refers to the detection of abusive language and actions in any text-based communication on digital platforms, including racism, sexism, hate speech, and other offensive behaviors (Pelle et al., 2018).

As more user-generated content is delivered across social media, the data becomes too massive for manual filtering. This information overload on the internet requires intelligent systems that can identify potential risks automatically. Naturally, there is a great demand for reliable automated systems to identify abusive language.

While research on offensive text identification has been extensively explored in English, it has received less attention in the field of Arabic language. The following summarizes some of previous studies on detecting offensive text in Arabic language.

In 2015, Al-Ayyoub et al., combined linguistic and machine learning features, such as part-of-speech (POS) tagging, sentiment analysis, and n-grams, to train a support vector machine (SVM) classifier. Experimental results showed a high degree of success in detecting offensive content on Arabic social media.

Alakrot et al. (2018), collected and experimented with a large dataset of YouTube comments in Arabic which contains a broad range of both offensive and inoffensive comments. The authors used this dataset to train a Support Vector Machine (SVM) classifier and experimented with combinations of word-level features, N-gram features and a variety of pre-processing techniques. Compared to classifiers reported by previous studies on Arabic text the proposed classifier achieves 90.05% accuracy.

Mohaouchane et al. (2019), trained and tested four different neural network architectures on a labeled dataset of Arabic YouTube comments. The performance of Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (Bi-LSTM), Bi-LSTM with attention mechanism, and a combined CNN-LSTM architecture was evaluated on the YouTube comments. The experimental results showed that the CNN-LSTM achieves the highest recall (83.46%), followed by the CNN (82.24%), the Bi-LSTM with attention (81.51%) and the Bi-LSTM (80.97%).

Husain (2020) proposed an ensemble machine learning approach for offensive language detection on Arabic language. The authors trained a number of ensemble machine learning classifiers, among the trained ensemble machine learning classifiers, bagging performs the best in offensive language detection with F1 score of 88%, which exceeds the score obtained by the best single learner classifier by 6%.

A study conducted by Omar et al.,(2020) executed a Comparative Performance of Machine Learning and Deep Learning Algorithms for Arabic Hate Speech Detection in OSNs by constructing a standard Arabic dataset that can be applied for the detection of hate speech and abuse. While previous datasets were only gathered from one social network, the suggested dataset includes YouTube, Twitter, Instagram, Facebook, and other networks. Twelve machine learning methods and two deep learning architectures were employed to confirm that the suggested datasets were effective. With an accuracy of 98.7%, Recurrent Neural Network (RNN) showed better results than other classifiers.

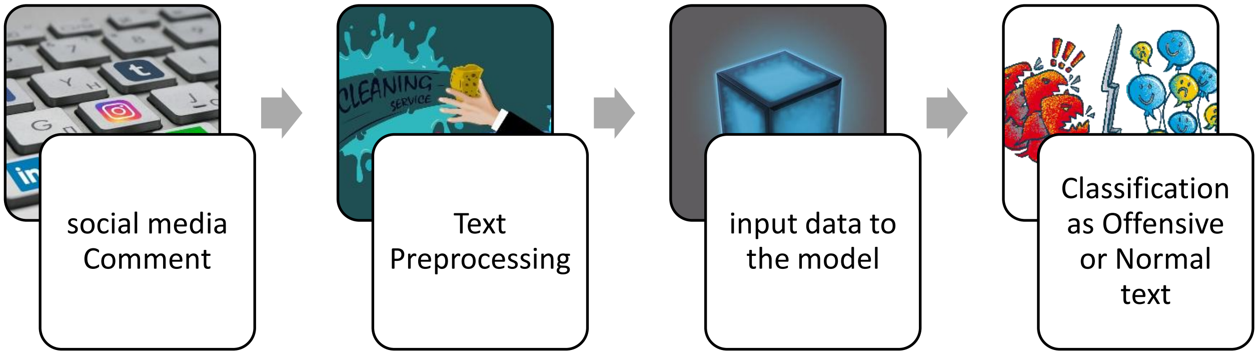
Others researchers discussed the topic using clustering techniques, Kanan et al.,(2021) proposed the use of clustering techniques to find online harassment and bullying. Several clustering techniques, such as K-Means and Expectation Maximization (EM), were used. In addition, they employed a variety of NLP technologies to achieve this goal. The findings show that in all of the tests, K-Means' training time is much less than EM's. Based on the variance in the applied NLP settings, the two clustering techniques performed differently in terms of accuracy.

Boulouard et al., (2022) explore the possibilities offered by BERT models to detect hateful and offensive speech in social media comments that are written in standard Arabic, or in any of the top three most spoken dialects in the Middle East (Gulf, Egyptian, and Iraqi). The dataset used contains hateful YouTube comments written in the aforementioned dialects. The authors trained different BERT-based models such as multilingual BERT (mBERTAR), AraBERT and multilingual (mBERTEN). Results showed that BERTEN has the best results with 98% accuracy, followed by AraBERT with 96% accuracy.

Aljuhani et al., (2022) in their research develop an effective detection model by employing deep learning and semantic and contextual features. This contribution proposed a deep learning approach that utilizes BiLSTM model and domain-specific word embedding's extracted from an Arabic offensive dataset. The approach was evaluated on an Arabic dataset collected from Twitter. Experiments showed that BiLSTM model achieves accuracy of 93%.

Methodology:

The aim of the project is to develop an accurate Offensive or Normal Text Classifier that can effectively distinguish between offensive and normal social media text content using Machine Learning Algorithms.



social media Comment

Text Preprocessing

input data to the model

Classification as Offensive or Normal text

- **Data Collection and Preparation:**
  The dataset comprises 9846 observations with 2 variables: "Comment" and "Target." The "Comment" variable represents Arabic text collected from various social media platforms, while the "Target" variable indicates the type of the corresponding text, which can be classified as either "normal" or "offensive." The project involved several steps to obtain the final model. These steps included:
  - **Data Preprocessing:**
    - removing duplicated values :

      number of samples Before :

Target	Comment
normal	6975
Offensive	2871

number of samples After:

Target	Comment
normal	6833
Offensive	2837

- convert the target column into numeric as follow:

Target	
normal	0
Offensive	1

- Remove non-Arabic characters.
- Remove extra white spaces.
- Remove diacritical marks(Arabic vowel marks) from the text.
- Remove towel character which is a character used to elongate Arabic letters.
- Remove new lines.
- Remove URLs by matching and replacing them with an empty string.
- Remove the Arabic conjunction "و" (waw) if it is followed by a word boundary.
- Remove numeric digits.
- Remove emoji characters and other Unicode characters outside the specified ranges.
- Removes all punctuation.
- Remove Arabic stop words.

Example:

Original text: "هذا نص عربي 123 ABC !@#", "مرحبا بك في العالم"

After text preprocessing: مرحبا العالم نص عربي

- Convert text to numeric using TF-IDF (Term Frequency-Inverse Document Frequency) which is an algorithm that uses the frequency of words to determine how relevant those words are to a given text.

- **Exploratory Data Analysis:** We employed term frequency analysis to identify the most commonly occurring words in each class text.

- **Model Selection**: We use two popular models, Support Vector Machines (SVM) and Extreme Gradient Boosting (XGBoost), for our machine learning tasks. Support Vector Machines are a powerful class of algorithms used for classification. They work by finding an optimal hyperplane that separates data points into different classes or predicts continuous values.On the other hand, XGBoost is an ensemble learning method that has gained significant popularity due to its excellent performance in a wide range of machine learning problems. XGBoost combines the strengths of gradient boosting with some additional enhancements, such as regularization, to improve both accuracy and speed. It iteratively builds a strong predictive model by adding weak models (decision trees) to the ensemble, each correcting the mistakes of the previous models.
- **Evaluation:** To evaluate the performance of the models, We applied the testing data to the models and compared the predictions with the actual targets using the following metrics:

- **Accuracy:** is defined as the percentage of correct predictions for the test data.
- **Precision:** is defined as the fraction of relevant examples (true positives) among all the examples which were predicted to belong in a certain class.
- **Recall:** is defined as the fraction of examples that were predicted to belong to a class with respect to all the examples that truly belong in the class.
- **F-score:** is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall.

Implementation:

For the implementation of our project, we selected Google Drive as our storage service provider. Google Drive offers a reliable and user-friendly cloud storage solution with vast storage capacity for the used dataset and project files. With its integration with other Google services, it provides easy accessibility and sharing capabilities. Additionally, Google Drive offers robust security features, ensuring the safety and privacy of our data. However, one drawback of Google Drive is its limited free storage capacity, which may require additional paid storage for larger datasets. Despite this limitation, we chose Google Drive for its overall reliability, accessibility, and security, making it an ideal choice for our project's storage needs.

To process and analyze our data, we chose Google Colab as our processing service provider. Colab, short for Google Colaboratory, is a cloud-based Jupyter notebook environment that provides free access to powerful GPUs and TPUs. This allows us to efficiently train and test our machine learning models on large-scale datasets. Colab offers good integration with Google Drive, simplifying the data loading process. Additionally, it provides a wide range of pre-installed libraries and frameworks, such as TensorFlow and PyTorch, facilitating the development and implementation of our models. One potential drawback of Colab is the limitation on session duration and available system resources for free users. However, given its extensive capabilities, free access to GPUs/TPUs, and easy collaboration features, we found Colab to be a suitable choice for our project's processing requirements.

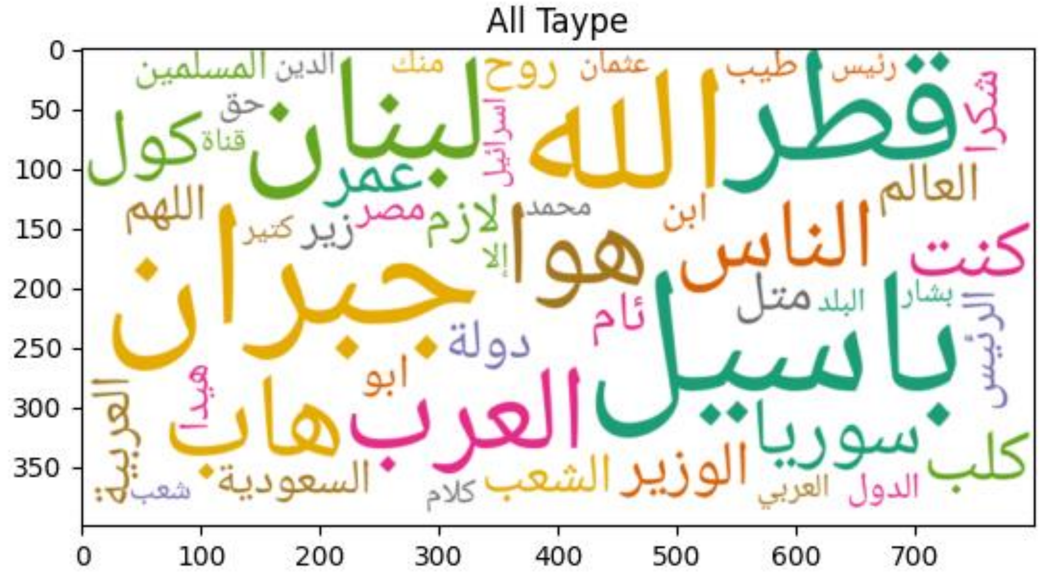
Results

Exploration Data Analysis :

We looked at the most frequent words in each class sentences and this is the results :

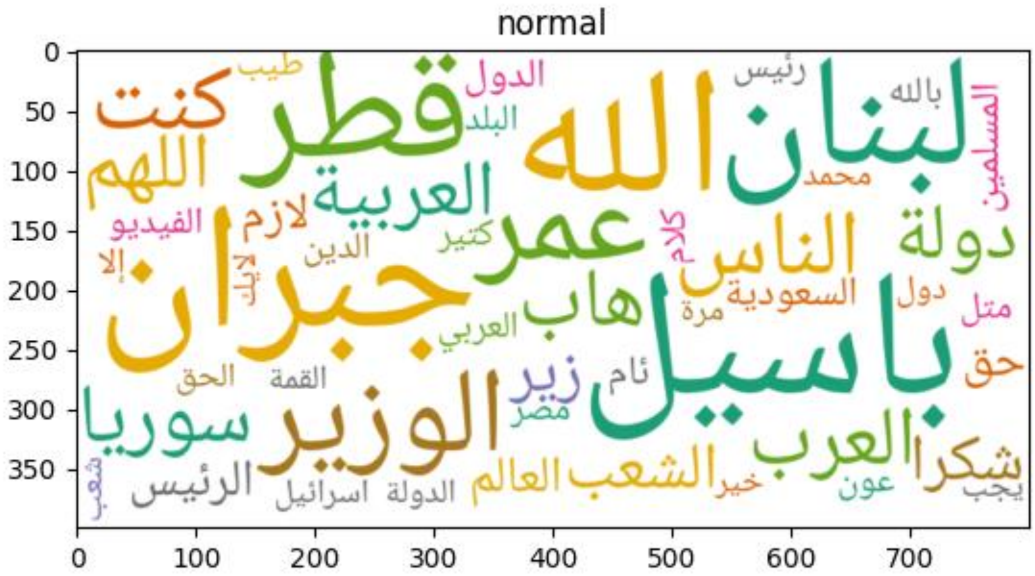
1) All class:

[('الله', 1563), ('جبران', 962), ('باسيل', 954), ('قطر', 320), ('لبنان', 292), ('العرب', 216), ('هوا', 216),
('هاب', 207), ('الناس', 203), ('سوريا', 203), ('كول', 193), ('كنت', 191), ('عمر', 181), ('كلب', 174),
('الوزير', 167), ('العالم', 164), ('العربية', 161), ('الشعب', 157), ('دولة', 157), ('متل', 146), ('لازم', 144),
('ثام', 138), ('روح', 136), ('اللهم', 134), ('السعودية', 129), ('الرئيس', 129), ('زير', 128), ('شكرا', 127),
('ابو', 127), ('المسلمين', 120), ('مصر', 117), ('ابن', 117), ('هيدا', 115), ('طيب', 111), ('الدول', 109),
('إلا', 109), ('حق', 107), ('منك', 104), ('العربي', 102), ('بشار', 101), ('البلد', 101), ('شعب', 101), ('قناة',
100), ('كتير', 100), ('رئيس', 99), ('كلام', 98), ('اسرائيل', 97), ('الدين', 95), ('عثمان', 94), ('محمد', 92)]



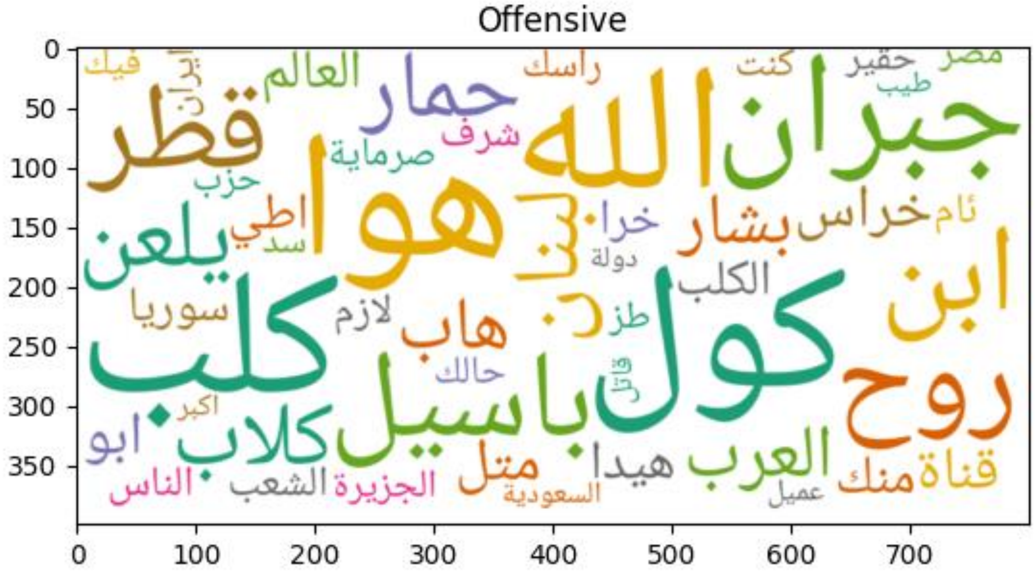
2) Normal class:

[('الله', 1195), ('جبران', 835), ('باسيل', 829), ('لبنان', 218), ('قطر', 213), ('عمر', 169), ('الوزير', 157),
('الناس', 155), ('كنت', 154), ('سوريا', 154), ('العرب', 147), ('هاب', 138), ('العربية', 136), ('دولة', 125),
('اللهم', 116), ('شكرا', 116), ('الشعب', 113), ('زير', 112), ('الرئيس', 111), ('العالم', 110), ('لازم', 99),
('حق', 96), ('السعودية', 93), ('الدول', 92), ('ثام', 92), ('المسلمين', 89), ('الفيديو', 85), ('العربي', 85), ('متل',
83), ('إلا', 81), ('كلام', 81), ('عون', 81), ('كتير', 80), ('رئيس', 80), ('طيب', 78), ('الدين', 78), ('البلد', 78),
('مصر', 77), ('بالله', 76), ('اسرائيل', 76), ('الحق', 75), ('محمد', 74), ('الدولة', 72), ('شعب', 72), ('القمة',
72), ('دول', 71), ('يجب', 70), ('مرة', 70), ('لايك', 70), ('خير', 69)]



3) Offensive class:

[('الله', 368), ('هوا', 198), ('كول', 189), ('كلب', 169), ('جبران', 127), ('باسيل', 125), ('قطر', 107), ('ابن',
86), ('روح', 84), ('يلعن', 82), ('حمار', 80), ('لبنان', 74), ('كلاب', 69), ('العرب', 69), ('هاب', 69), ('بشار',
67), ('خراس', 66), ('ابو', 65), ('متل', 63), ('قناة', 59), ('العالم', 54), ('منك', 54), ('هيدا', 53), ('اطي', 51),
('سوريا', 49), ('خرا', 49), ('الكلب', 48), ('الناس', 48), ('شرف', 46), ('ثام', 46), ('لازم', 45), ('الشعب', 44),
('صرماية', 44), ('راسك', 42), ('الجزيرة', 40), ('مصر', 40), ('حقير', 40), ('فيك', 39), ('ايران', 39), ('طز', 39),
('حزب', 38), ('حالك', 38), ('سد', 38), ('كنت', 37), ('السعودية', 36), ('عميل', 35), ('قاتل', 35), ('طيب',
33), ('اكبر', 33), ('دولة', 32)]

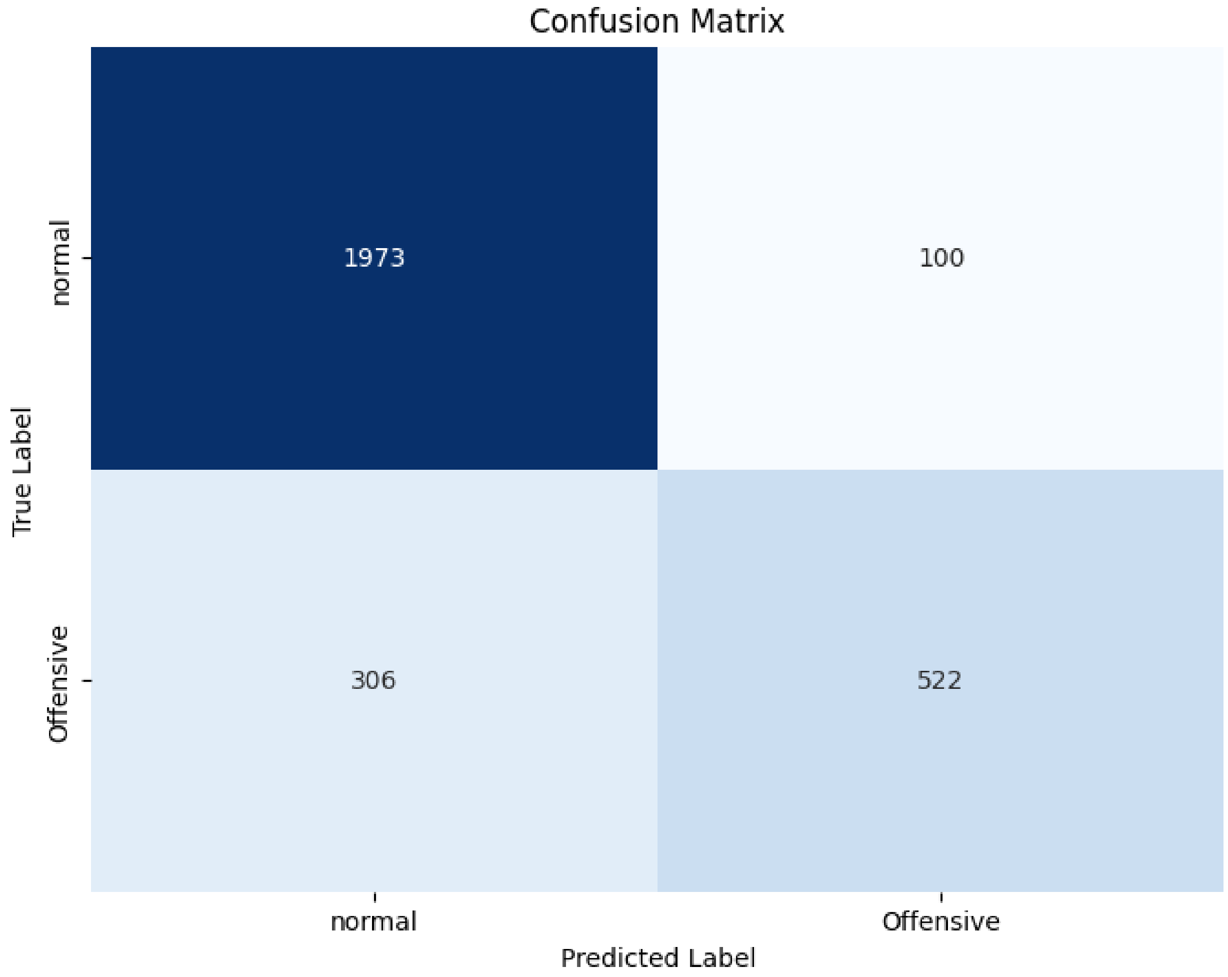


Model Evaluation:

To check the performance of the models we applied the testing data to the models and compared the prediction with the real target (confusion matrix) and the result as below.

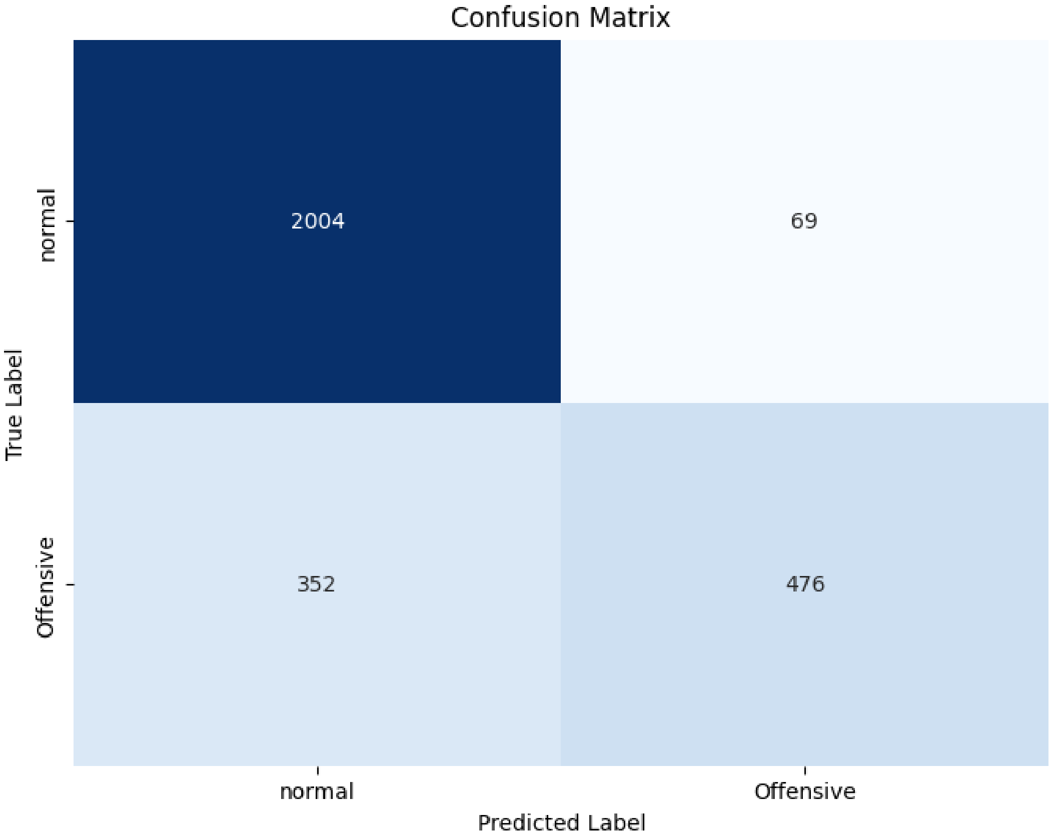
1) Support vector machines Model:

	precision	recall	f1-score
Normal	**0.87**	**0.95**	**0.91**
Offensive	**0.84**	**0.63**	**0.72**
Overall Accuracy			**0.86**



2) xgboost Model :

	precision	recall	f1-score
Normal	**0.85**	**0.97**	**0.90**
Offensive	**0.87**	**0.57**	**0.69**
Overall Accuracy			**0.85**



Conclusion

In conclusion, this project highlights the importance of addressing offensive text detection in the Arabic language. With the increasing use of Arabic in various online platforms, there is a pressing need for effective methods to filter and identify abusive language. Through the using the dataset consisting of Arabic social media text, we have developed a model that can accurately classify text as either "normal" or "offensive."

Throughout the project, we encountered several challenges, including the availability of labeled data and the complexity of Arabic language processing. However, by utilizing popular machine learning algorithms such as Support Vector Machines (SVM) and Extreme Gradient Boosting (XGBoost), we were able to achieve a high level of accuracy in offensive text detection. The use of cloud computing and tools like Google Drive and Colab greatly facilitated the handling of large-scale data and the training of our models. We used TF-IDF (Term Frequency-Inverse Document Frequency) in text preprocessing. TF-IDF helps us quantify the importance of each word in a document by considering both its frequency in the text and its rarity across the entire dataset.

This project has the potential to benefit a wide range of stakeholders. Social media platforms and online communities can leverage our model to automatically detect and filter offensive language, thereby creating a safer and more respectful online environment. Researchers in the field of Natural Language Processing (NLP) can also benefit from our work by gaining insights into effective techniques for offensive text detection in the Arabic language.

future work

For more improvements in the Arabic language offensive text detection field, future study should concentrate on two main areas.

The first step is to gather more varied and thorough data from different Arabic social media platforms. This larger dataset would increase the model's capability to recognize and categorize foul language in a variety of events and contexts.

Secondly, the development of the model can be extended by exploring advanced machine learning techniques and algorithms. The effectiveness of the model in detecting objectionable material may be improved by including deep learning techniques like recurrent neural networks (RNNs) or transformer models like BERT. These methods can lead to greater accuracy and precision and have shown encouraging results in challenges involving natural language processing.

References

Alakrot, A., Murray, L., & Nikolov, N. S. (2018). Towards accurate detection of offensive language in online communication in arabic. Procedia computer science, 142, 315-320.

Albirini, A. (2016), Modern Arabic sociolinguistics: Diglossia, variation, codeswitching, aitudes and identity, Routledge.

Aljuhani, K. O., Alyoubi, K. H., & Alotaibi, F. S. (2022). Detecting Arabic Offensive Language in Microblogs Using Domain-Specific Word Embeddings and Deep Learning. Tehnički glasnik, 16(3), 394-400.

Boulouard, Z., Ouaissa, M., Ouaissa, M., Krichen, M., Almutiq, M., & Gasmi, K. (2022). Detecting Hateful and Offensive Speech in Arabic Social Media Using Transfer Learning. Applied Sciences, 12(24), 12823.

Hmeidi, I., Al-Ayyoub, M., Abdulla, N. A., Almodawar, A. A., Abooraig, R., & Mahyoub, N. A. (2015). Automatic Arabic text categorization: A comprehensive comparative study. Journal of Information Science, 41(1), 114-124.

Husain, F. (2020). Arabic offensive language detection using machine learning and ensemble machine learning approaches. arXiv preprint arXiv:2005.08946.

Kanan, T., Kanaan, G. G., Al-Shalabi, R., & Aldaaja, A. (2021). Offensive Language Detection in Social Networks for Arabic Language Using Clustering Techniques. International Journal of Advances in Soft Computing & Its Applications, 13(2).

Mohaouchane, H., Mourhir, A., & Nikolov, N. S. (2019, October). Detecting offensive language on Arabic social media using deep learning. In 2019 sixth international conference on social networks analysis, management and security (SNAMS) (pp. 466-471). IEEE.

Omar, A., Mahmoud, T. M., & Abd-El-Hafeez, T. (2020). Comparative performance of machine learning and deep learning algorithms for Arabic hate speech detection in osns. In Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020) (pp. 247-257). Springer International Publishing.

Pelle, R., Alcântara, C., & Moreira, V. P. (2018, October). A classifier ensemble for offensive text detection. In Proceedings of the 24th Brazilian Symposium on Multimedia and the Web (pp. 237-243).