



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

To automate the process of detecting
inappropriate license plates in English
using an NLP model.

By
Abdulmalik Elghedafi

Supervisor: Mark Huntly

Project submitted as part of the requirements for
the award of
MSc Software Engineering (Web Development)

February 2026

DECLARATION

I, Abdulmalik Elghedafi declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed : Abdulmalik Elghedafi

Date : 02/02/2026

FORM OF CONSENT

I Abdulmalik Elghedafi hereby consent that my Project, submitted in candidature for the MSc Software Engineering (Web Development) degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed : Abdulmalik Elghedafi

Date : 02/02/2026

Abstract:

Refer to the MSc handbook for further information.

Add abstract here.

Table of Contents

Table of figures:	5
Table of tables:	6
Introduction	7
Research question	7
Project Aim	7
Objectives:	8
Motivation for this study	8
Literature review	9
Offensive language within license plates domain:	12
Sentiment Analysis within NLP:	14
Machine learning within NLP:	14
Word Embeddings:	17
Word2Vec:	18
Glove:	19
Fast Text:	20
Deep learning:	21
Natural Language Processing (NLP)	22
BERT	22
Roberta:	23
Deep learning:	25
Recurrent Neural Network (RNN):	25
Long Short Term Memory (LSTM):	26
Why was attention introduced?	27
Introduction to Transformers:	27
Encoder & Decoder:	28
Tokenization	21
Methodology	30
Results and Discussions	36
Discussion:	40
Conclusion and future work:	43
Bibliography	45
Appendix:	52
Project Proposal:	53
Project proposal:	58
Introduction	60
Hypothesis	61
Project Aim.....	61
Objectives:.....	62

Literature review	64
Arabizi	64
Natural Language Processing (NLP)	65
BERT	65
ARABERT	66
Offensive language in Arabic	66
Preprocessing and normalisation techniques	69
Dataset.....	72
Toolkit	73
Motivation for this study.....	73
Methodology	73
Gantt Chart	76
Project planning	79
Development methods.....	79
Dissertation	79
Ethical	79
Expected outcomes / deliverables.....	80
Deliverables	80
Resources	81
Conclusion	82
Bibliography	83

Table of figures:

Figure 1 - A visual graph of the transformer models performance [2].....	12
Figure 2 - An overview of the algorithms within Machine Learning [10].....	15
Figure 3 - Techniques used within word embedding [19]	18

Figure 4 - A 2D diagram presenting how words with a similar relationship are positioned [28].	20
Figure 5 - The LSTM model architecture [36]	26
Figure 6 - Analysis across models with metrics [46]	24
Figure 7 - A diagram representing the transformer architecture [50].	29
Figure 8 - BERT confusion matrix.	37
Figure 9 - RoBERTa confusion matrix.	38
Figure 10 - DistilBERT confusion matrix	39
Figure 11 - The research onion [64]	48

Table of tables:

Table 1 - Comparison of the models metrics [1]	10
Table 2 - Evaluation metrics of transformer models [2]	11
Table 3 - Results table for YOLOv8 and v5 model [3]	13
Table 4 - Results for BERT classification report	36
Table 5 - RoBERTa classification report	36
Table 6 - DistilBERT classification report	36

Introduction

This research investigates a solution to automate the process of detecting offensive vehicle number plates specifically within the DVLA to replace the current process of identifying offensive number plates, which is a manual process and is time consuming.

Research question

The research question is, which transformer-based model in Natural Language Processing (NLP) delivers the best performance between BERT, RoBERTa and DistilBERT in detecting inappropriate text-based vehicle registration number (VRN) on unseen data?

Project Aim

The aim of the project is to evaluate and compare three transformer models in detecting inappropriate DVLA vehicle registration numbers (VRN) written in English alphanumeric characters. The project will involve training the three NLP models on a training dataset and testing them on unseen data to output quantifiable data.

Objectives:

1. To critically evaluate the literature to understand the state of the field within NLP and vehicle registration numbers.
2. To prepare and preprocess a labelled dataset to be trained on the model during the first week of the implementation process.
3. To evaluate how well BERT, RoBERTa and DistilBERT will perform on unseen data using quantifiable metrics.
4. To evaluate the models based on the metrics chosen.

Motivation for this study

The motivation for this study is to address a gap in the manual detection of offensive and non-offensive vehicle registration number plates, where the detection of curating a list of suppressed plates is done through a manual process by a small team of DVLA staff. This process is both time consuming and prone to mistakes, where offensive plates can slip through the net if not thought out well by the staff. Therefore, an automated process has been considered through a curiosity that could lead to a need within the DVLA to replace the manual process with an automated version instead.

Literature review

Given the rise of social media and having unrestricted access, where users can freely share their thoughts and emotions, it can also introduce the use of offensive and hate language being sent online. This can include, but not limited to treating others unfairly, trigger violent behaviour, creating a harmful setting online and cyberbullying. In this study, the author investigates the performance on a BERT transformer model in identifying offensive content through online discussions and explores how effective it can be at determining hate speech online [1]. In this study, they use a fine-tuned BERT model in detecting offensive and hate speech content.

A strength outlined in the study, has shown that BERT had achieved a 90% accuracy rate as shown in figure 1. This highlights its ability in handling different categories such as detecting offensive language [1]. When both precision and recall are well-aligned with each other, it indicates the model can detect offensive and hate speech without favouring one model over the other by false flagging offensive comments. As shown in the below table, other deep learning models were also evaluated as part of this study, however BERT demonstrated superior performance among all the models tested in the study.

Algorithm	Accuracy	Precision	Recall	F1-Score
BERT	0.90	0.90	0.90	0.89
LSTM	0.85	0.84	0.85	0.85
Bi-LSTM	0.58	0.60	0.58	0.59

GRU	0.85	0.84	0.85	0.84
RNN	0.84	0.84	0.85	0.84

Table 1 - Comparison of the models metrics [1]

However, a limitation of BERT within the study, suggests that it heavily relied on the model requiring lots of labelled data, which can be difficult to find and are not balanced [1]. Future research suggests looking into semi-supervised learning, therefore removing the necessity of manually labelling data within the dataset, and letting the model naturally learn patterns to understand how to improve and reducing manual labour.

A further concern highlighted is that BERT may underperform due to detecting new or subtle types of hate speech due to the data not being labelled correctly in the training dataset. This can lead to the model making mistakes such as false flagging words such as hate speech (false positive) or fail to predict hate speech by presenting it as non-offensive (false negative) [1]. This suggests that the results of the study could be represented incorrectly due to biased results of the model's data being trained on.

In contrast, Chakarverti et al. [2] investigates the use of transformer based models in tackling hate speech, by evaluating the performance based on metrics. The study shows the use of a referenced dataset containing 24,783 labelled entries and split into categories such as offensive language, hate speech or neither [2]. However Arif and Nayak [1] use the same amount of entries in their dataset sourced from Kaggle, however it was unverified due to it being unreferenced [1]. This highlights a gap in transparency and reproducibility within their study. A further observation by Chakarverti et al. [2] is that they used a dataset split, which provides a clearer picture about the approach they used [2].

In contrast to Arif and Nayak [1] who did not describe their dataset split, Chakarverti et al. [2] demonstrated that the data was evaluated fairly and was robust in their testing phase.

While Arif and Nayak [1] did not include Roberta as part of the transformer models in their study, Chakarverti et al. [2] performed a comparative evaluation against models such as Albert, Bart, Bert, DistilBert and Roberta. Table 2 provides a detailed view of the metrics among all the models, while figure 1 illustrates a visual representation of how the models performed. This shows a wide range of transformer models in detecting hate speech. In their study, Roberta outperforms BERT, with an accuracy and recall rate of 0.89%, precision and F1 rate of 0.88% [2]. This suggests that the Roberta model is better tasked at handling hate speech and offensive language detection than Bert.

Model	Accuracy	Precision	Recall	F1	F1.5
ALBERT	0.8707	0.8632	0.8707	0.8598	0.8632
BART	0.8435	0.8381	0.8435	0.8387	0.8401
BERT	0.8812	0.8795	0.8912	0.8815	0.8847
DISTILBERT	0.8836	0.8550	0.8639	0.8555	0.8581
ROBERTA	0.8912	0.8868	0.8912	0.8859	0.8874
MA	0.8701	0.8645	0.8721	0.8643	0.8667
WA	0.8802	0.8746	0.8800	0.8742	0.8566

Table 2 - Evaluation metrics of transformer models [2]

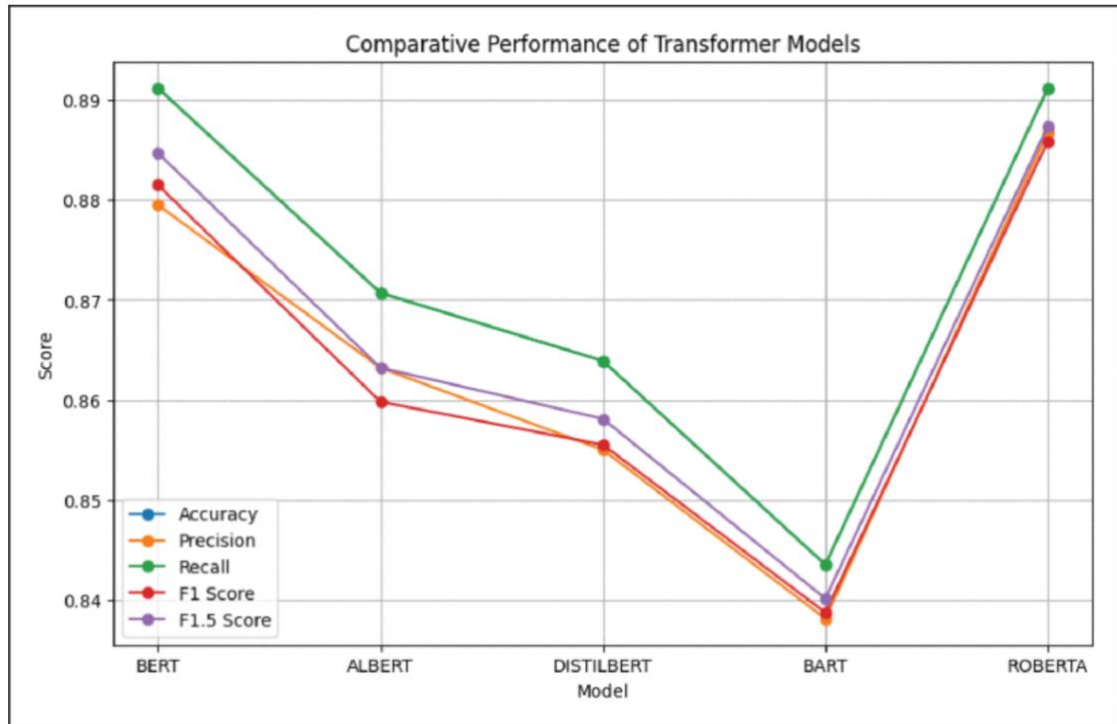


Figure 1 - A visual graph of the transformer models performance [2]

While the study looked into using transformer models in detecting hate speech detection within social media. It did not investigate formatted text such as vehicle registration plates that use alphanumeric to structure the text. This is a gap that is presented in this study, and this project looks into developing a solution to address the issue.

Offensive language within license plates domain:

In this paper, the author presents an issue with using manual monitoring systems due to the likelihood of encountering mistakes and being labour intensive. A proposition for the use of an automated system, has been announced to solve such issues. However, the study states that the current License Plate Recognition (LPR) system, which allows the identification of vehicle registration plates (VRN), presents drawbacks. These include capturing VRN's during low lighting, diverse VRN plates, blur caused by fast moving vehicles [3].

The author highlights a viable solution by improving upon the YOLOv8 model based on the Automated Number Plate Recognition (ANPR) system of detecting objects. This fixes the issues mentioned previously about the LPR system and therefore the new solution can lead to improved accuracy and processing times, which can be useful for real world scenarios [3].

The author uses a custom version of YOLOv8 that integrates on the Automatic License Plate Recognition (ALPR) system, which improves the identification of license plates and precision. As shown in results table 3 [3]. The YOLOv8 had greatly outperformed the YOLOv5 model. However, it has been noted that the system does not support multiple languages i.e. within different regions Another limitation is the results of the Optical Character Recognition (OCR) to show how well it could read a plate and see if it makes any false positives with character readings with letters that read similar to numbers i.e. “1” and the letter “l” or 0 and O [3].

Parameters	Custom Trained YOLOv8 (Ours)	Custom Trained YOLOv5
mAP50 (Accuracy)	96.2%	78.2%
mAP50-90	75.4%	60.1%
Precision	95.5%	70.6%
Recall	91.0%	68.5%
F1-Score	94.3%	69.2%

Table 3 - Results table for YOLOv8 and v5 model [3]

As opposed to Arora et al. [4] which explores the use of ANPR using YOLOv9 and OCR to detect license plates. A strength outlined by Arora et al. is the transparency of details about the dataset split for training, validation and test, which provides the ability to reproduce the study and removes bias from the results [4]. Unlike in Kasab et al. [3] paper, which did not reference the dataset and neither did they reference the data split. This raise concerns on whether the results based on evaluated metrics have been inflated

without proper referencing and a detailed view on how the model's dataset has been split.

Due to many studies that use ANPR, which are based on the implementation of Convolutional Neural Network (CNN) and use models such as YOLO and OCR to detect and read Vehicle Registration Numbers (VRN). There hasn't been a study that focuses on the use of Natural Language Processing (NLP) models to detect VRN plates. For this reason, it is worth investigating this area of NLP transformer models for real world environments.

Sentiment Analysis within NLP:

Sentiment Analysis is a core area within NLP, where text is analysed through areas such as social media, online reviews, user insights, advertising and it identifies whether the tone of the text is positive, neutral or negative [5] [6]. This is based on the authors approach based on the category chosen [7]. Sentiment Analysis, can help computers to understand the author feels on each topic for example online reviews [5].

Machine learning within NLP:

Machine learning (ML) is a subset of Artificial Intelligence (AI) which provides a system that can run automatically and allows the software to learn on its own through trial and error without interacting with code written by a human to provide its instructions [8]. ML uses data to strengthen the decision making by training models [9]. It does this by analysing the data to identify patterns to enable the system to make accurate predictions over an extended period of time without requiring manual input [8]. ML has four techniques to learn from data: reinforcement, supervised, semi-supervised and unsupervised methods, as highlighted in figure two [10].

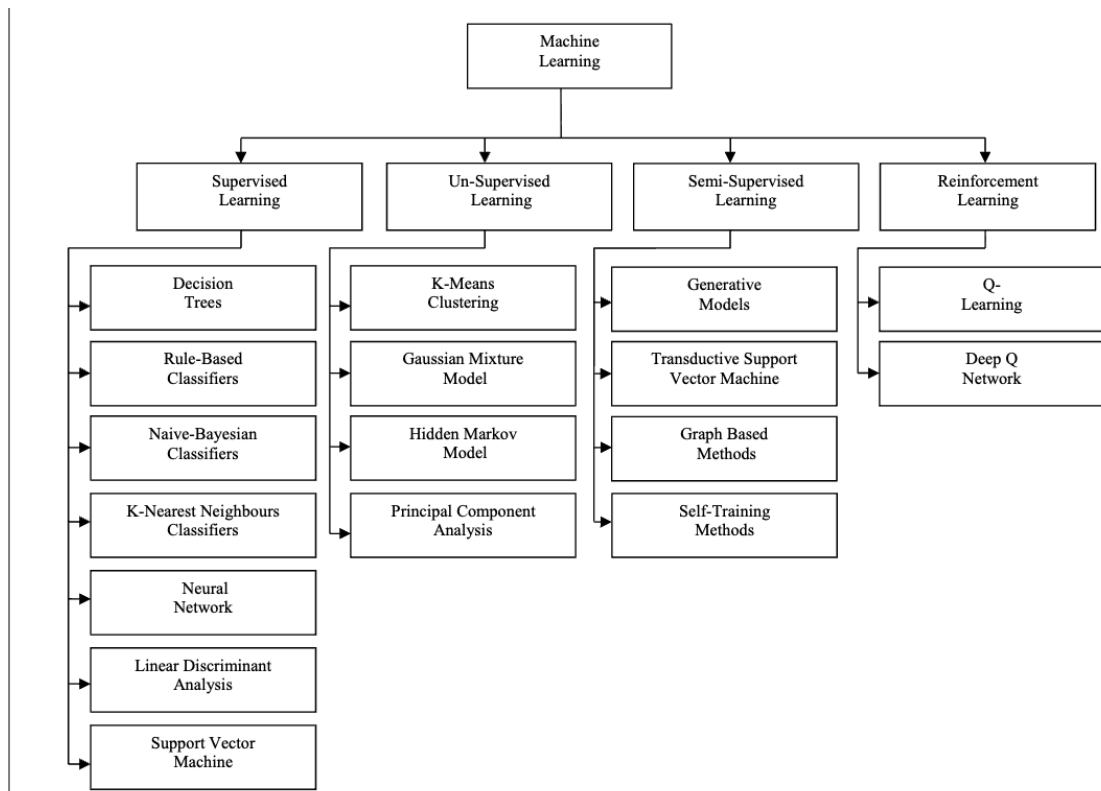


Figure 2 - An overview of the algorithms within Machine Learning [10]

Supervised machine learning is a technique in ML where the algorithm is trained on labelled data/datasets known as the input and this is mapped to the correct output [11]. The aim within supervised learning is to derive a model that can generalise on labelled trained data to handle unseen data [10].

Classification is a task within supervised learning where the aim is to predict the correct category/label of the data. This is important to ensure the model can make correct predictions and find patterns based on the input, which can then be trained on unseen data [12].

Supervised learning is relevant for the project due to working with labelled data to detect offensive unseen data formulated in a license plate. This is important, as the model can learn different patterns and make accurate future predictions.

Unsupervised learning is knowing the input data without the knowing the correct output. This forces the model to learn on its own through unlabelled datasets and identify hidden patterns independently to discover the correct output [13]. Due to the project, the dataset will use labelled data, that requires a model to predict offensive/non-offensive text using evaluation metrics. Therefore, the unsupervised method is less relevant in this case.

Furthermore, there are two other learning models, reinforcement and semi-supervised learning. The reinforcement learning (RL) method is the process where an agent will engage within an environment through trial and error to acquire the maximum rewards. RL does not use labelled data as the input, but the model will be alerted once the expected result is fulfilled [14]. Semi-supervised learning is the process of using a small quantity of labelled data and larger quantity of unlabelled data in order for the algorithm to develop a model to categorise data [10].

In this study, Richardson et al. [15] focuses on applying semi supervised machine learning to detect hate speech that has been posted online i.e. social media platforms. They used Support Vector Machine (SVM) as the chosen predictive model for the study. A strength outlined in the study shows that the semi-supervised model can be almost as successful as the supervised model when using a substantially reduced dataset and still provide optimal results. Therefore demonstrating the reduction of time spent on preparing and labelling data, due to using a smaller dataset, which will yield similar/improved results [15]. As a result, this approach can reduce time spent on labelling data and costs involved when working on real world projects.

Despite the strengths highlighted in the approach of the study, they both face limitations that could affect their real-world usage. A significant weakness, highlighted in Richardson's et al. [15] study, was that the model would often label data incorrectly, that contained bad language and label it as hate speech. This implies the model lacks the ability to differentiate between hate speech, neutral and defensive language. Therefore, impacting the accuracy and precision rate of the results within a dataset [15].

Word Embeddings:

Word embeddings in NLP, is a technique used that helps obtain how comparable words are based on their definition and relationships. A word embedding works by converting each word into a numerical vector [17]. In word embeddings, models are trained by analysing a substantial volume of text and modify the word vectors based on how words appear given their context [18]. In figure 2 [19], presents the techniques used in word embedding.

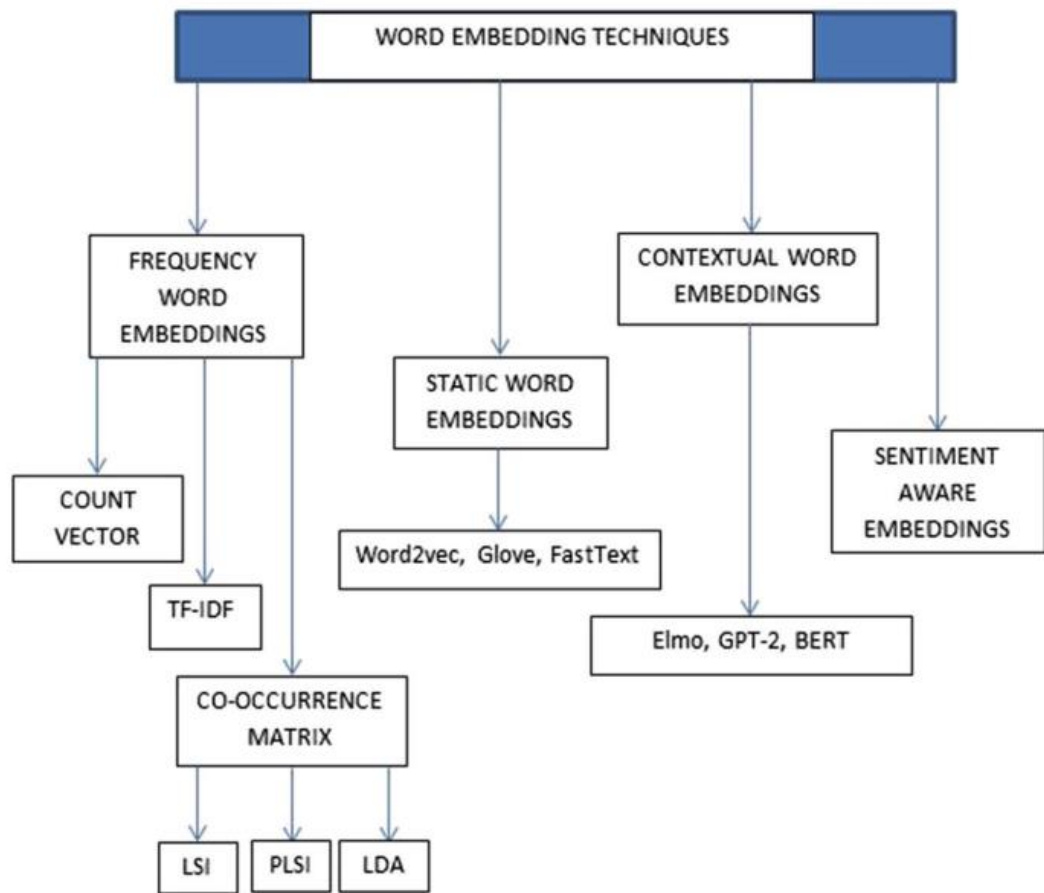


Figure 3 - Techniques used within word embedding [19]

Bengio et al, first introduced the concept of utilising vectors, represented by numerical figures, who employed the idea of using a feedforward neural network model aimed at predicting words. To which, Mikolov et al. then built on top of the idea, by implementing models such as skip gram, word2vec and continuous bag of words which also employs the use of predicting words [17].

Word2Vec:

Word2Vec is a model employed in word embeddings, which is created to illustrate how words with a similar definition are placed next to each other in a vector area. Word2Vec is used to obtain the relationship and context of words and their positioning within a sentence [20].

The advantages of Word2Vec is that it can be trained rapidly and at an efficient rate when working with significantly sized datasets. However the performance is based on the sampling rate, dimension of vector and selected model for the experiment [21]. That said, a limitation of Word2Vec is that it does not understand context between words, when compared with transformer models such as BERT [22]. Another advantage is that Word2Vec provides the use of a simple neural network to train word embeddings, by predicting word context between words and their context by linking words that share a similar context [23].

Despite its benefits, Word2Vec is constrained by embeddings which cannot be shared among other languages, therefore for every language, a new model is required to be trained from scratch [19].

Glove:

Global Vectors for Word Representation, known as GloVe, is a method used within NLP for un-supervised learning [24] . It was first introduced by Pennington et al. [25] in 2014. It was developed to obtain the definition of words by producing numerical vectors for words [24] [26]. Unlike Word2Vec, which looks at the local context of a sentence, Glove works by looking at the entire dataset to consider a global approach to understand the relationship between words and portrays each word by mapping it to a numerical vector [27]. As shown in figure 3 [28], it showcases how words are represented in a linear vector framework. This displays how words with similar relationships based on context are positioned near each other.

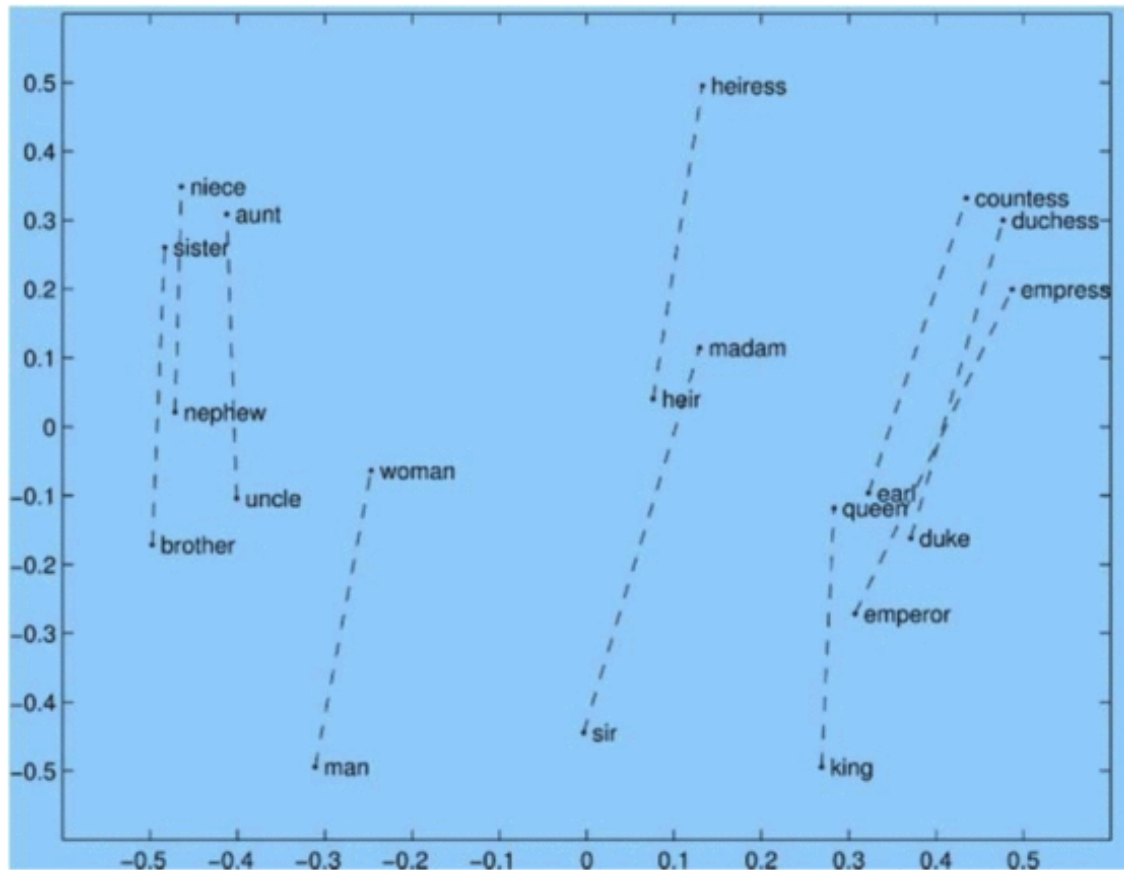


Figure 4 - A 2D diagram presenting how words with a similar relationship are positioned [28].

Fast Text:

FastText is a model, that expands on the original Word2Vec word embedding, that was developed by Mikolov et al. The FastText model uses a skip gram model, separating each word into smaller chunks of character n-grams and then a vector is created based on each chunk of characters formed from the word. FastText learns by looking at the surrounding words to form the context of words. That said, an advantage over Word2Vec is that FastText is able to provide a word embedding for words that it has not trained on, uncommon words and incorrectly spelled words due to using character-based tokenization [29].

The author, Soto et al. [30] presents a study of looking into the detection of hate speech, by investigating the use of four word embedding models known as Glove, FastText,

Word2Vec and Wang2Vec. The paper investigates comparing if word embedding that is trained on big datasets are required to provide strong levels of accuracy, compared to word embeddings that are trained on smaller datasets along with lower processing time.

In the methodology section, the author states, that they evaluated four-word embeddings: Word2Vec, Glove, FastText and Wang2Vec. That said, the authors kept the classification model: CNN the exact same for each experiment they conducted for each word embedding, therefore reducing bias and ensuring consistency across the results.

Tokenization

Tokenization is a critical technique used within NLP where text is separated into tokens. A token is the process where data is broken down into smaller chunks of data. The data can include a list of words, characters or even a sentence [38] [53]. There are three area's that tokenization can fall under. The first approach is word level, where text is separated using whitespace or punctuation. While this method can be simple to implement and used for NLP cases, Omar and Muccini outline a limitation where any word not included within the model's vocabulary will be ignored [54]. Additionally, if the model uses a large dataset to contain a wide range of words, it can lead to increased memory consumption and slower performance of the model [54]. The second approach is character level, where the model separates the text into a singular character. The author states that this method avoids the issues seen at word-level as each character is known to the model, therefore performing well on unseen words. That said, a negative implication shows that it can generate higher token counts which can lead to increased processing time to train data, which can lead to using increased energy usage compared to word and subword level [54]. The final approach is sub word level, where words are broken down further into frequent parts that appear often. Strategies like Byte Pair

Encoding (BPE) and Sentence Piece construct tokens, using smaller units, allowing the model to have enhanced performance and handle challenges with unseen words [54].

Natural Language Processing (NLP)

Natural Language Processing (NLP) is a sophisticated discipline where artificial intelligence (AI), software engineering and semantics are combined into one. NLP explores how the human language and computers can work together to listen, process and understand text. NLP can then generate text in a way that mimics the human language naturally. This is important as it allows people to communicate in a way that is normal, without requiring to learn or remember special commands for the computer to understand [38] [39]. An example of NLP are virtual assistances and chatbots, such as Alexa by Amazon [40], Siri by Apple [41], and Google assistant [42]. These are widely powered virtual assistants powered by AI, which consumers interact with everyday [38].

BERT

The **B**idirectional **E**ncoder **R**epresentation from **T**ransformer known as BERT is a potent language model developed by Google. BERT relies on the transformer architecture and specifically uses the Encoder part [43]. BERT scans text in both directions within a sentence and focuses on the context of each word in relation to adjacent words, to better understand its meaning [44]. BERT is trained based on two stages. The first stage involves pre-training the model based on large datasets such as Wikipedia and BookCorpus [44] [43] [45]. Then it uses fine-tuning to be used on particular tasks [44].

Roberta:

RoBERTa, is a transformer-based model, which has delivered superior performance and outperformed previous models. It is a text classification model that has been acknowledged in its capability of understanding the context in a sentence to be able to detect whether it is offensive or not [46].

In this paper, the author looks at the use of RoBERTA and LangChain to discover whether text presented in this study is offensive [46].

A strength outlined in the study, presents a large dataset which includes 24,783 distinct labelled data [46]. This is important as the model can train on unique data and identify different patterns to enhance how well the model performed to predict whether the word is offensive or not. By using contextual awareness throughout the tweets, the model can also dictate if the tweet is offensive, without relying on specific character patterns within the text.

That said, a limitation is that the dataset is extremely imbalanced, where 77.4% of data is offensive and 16.8% is hateful [46]. This introduces bias towards the model during the training phase, as it will learn and predict more offensive data, as it will frequently appear more than hateful data. This is important because the model will not have enough data to train on for the hateful category, therefore making poor predictions and missing out on hateful data during generalizability, which can lead to poor metrics.

A further strength provided within the methodology section; is the transparency of how the experiment was achieved. The authors detail each step of the process from transforming all text to lowercase, removing whitespace to the tokenization of each

tweet using RoBERTa and the sequence length of 128 tokens for the truncation. They also mentioned the training process taken for transformers through the Hugging Face library and using PyTorch as the backend. By stating an 80-20 split for the dataset [46], this reinforces the steps they've taken to produce their experiment, which offers a fair comparison of how they've conducted their research by being transparent and following the usual approach of conducting such study. Therefore, strengthening the credibility of the study.

A strength outlined in the results, as shown in figure 6 [46] highlight, that RoBERTa achieves outstanding performance of 90%+ for each metric, compared to SVM and BERT, therefore demonstrating how accurate RoBERTa can be at detecting offensive text. A high performance by the model, can indicate how well the model can generalise across unseen data, by having a good amount of training, therefore reducing the model from overfitting. As well as making less mistakes.

Model	Accuracy	Precision	Recall	F1 Score
SVM (Baseline)	75%	73%	76%	74%
BERT-Based Classifier	87%	85%	86%	85.5%
Proposed RoBERTa	91%	90%	91%	91%

Figure 5 - Analysis across models with metrics [46]

The research gap that the project will look into is the use of evaluation metrics by evaluating the use of analysis on how well the model predicts offensive vehicle registration plates through the use of visualisation graphs, which will investigate showcasing the insights of the model's ability to generalise on unseen data, prior to learning on the training dataset.

Deep learning:

The core concept of sequence to sequence (Seq2Seq) modelling is that a piece of text is inserted into the model as input, which creates a sequence of output. It was created originally for machine translation. The models input, would act as the encoder, which renders the context and passes it off to the decoder [31].

Recurrent Neural Network (RNN):

A Recurrent Neural Network (RNNs) is considered a neural network model, that is designed to identify patterns of data within sequences. The examples of data can be numerical, text or handwritten data and these are frequently derived from real world environments such as trading markets and sensor-based networks. However, RNN's are widely used within a practical approach, and can be utilized within multimedia tagging, generating images with text, voice recognition and generating text within language modelling [32].

RNN works by processing the text from left to right, by sharing identical parameters such as weights and activation function across all the steps. The hidden layer between the input and output, will remember the word from the previous step within the hidden layer, to make an accurate prediction for the next layer [33].

That said, a limitation of RNNs is that it remembers the recent steps, but loses the meaning of long term memory, therefore it becomes difficult to make future predictions on long pieces of text due to the vanishing gradient problem on training data [34] .

Long Short Term Memory (LSTM):

Long Short Term Memory (LSTM) is well established RNN architectural model that was introduced as a solution to address the issues of vanishing and exploding gradients during the point of executing detailed pieces of text [35].

LSTMs utilize cells which are placed in the hidden layer of the neural network and helps to preserve and apply information throughout long sequences of data. As shown in figure 3 [36], the workings of the model are presented. There are three gates; input, output and forget gate, which are all managed by each cell. The input gate determines the level of new data that enters the cell. The output gate, dictates which information to keep or delete from the previous state and the output gate, decides what data will get passed to the next layer [37].

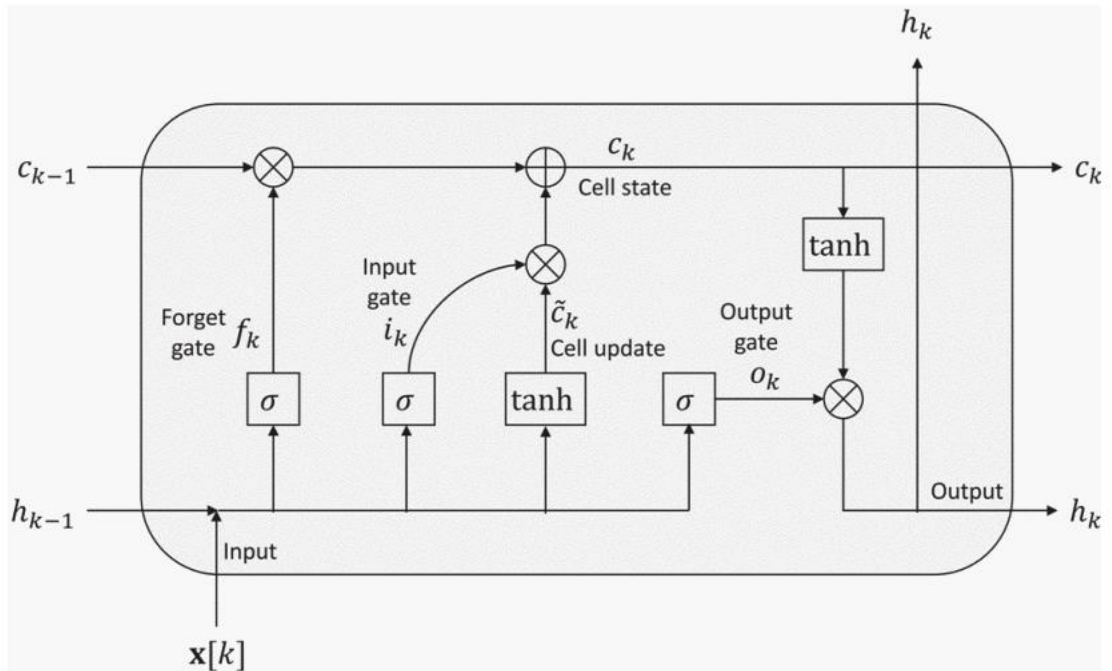


Figure 6 - The LSTM model architecture [36]

Why was attention introduced?

Attention was first introduced by Bahdanau et al. [47] to let computers assign processing power to allow focus on higher priority tasks, while reducing information overload when computational resources are limited. Attention helps organise important text within a sentence for the current task and decreases its attention on trivial data that is not in the scope of importance, therefore improving the efficiency of the system as well as accuracy [48].

In NLP, the machine translation uses the decoder, to process only the significant data from the input to the output being generated through using the attention model. This helps focus on parts of the sentence instead of the entire text. This helps the attention model provide an immediate pathway from the input to the output, therefore helping solve problems that recurrent neural networks had produced such as the vanishing gradient issue [48].

Attention later, become a well-known system in deep learning and to be used for activities such as sentiment analysis, text classification, speech recognition and image captioning etc. All models that had utilized the Attention model were regarded as state-of-the-art [47].

Introduction to Transformers:

The transformer architecture model, first revealed by Vaswani et al. [49] [50], in the paper published in 2017 named “Attention is All you Need” is a neural network that has changed the area of NLP. This has been done by improving how the neural network models can read and understand information within sentences due by understanding the context [49].

The transformer relies on using techniques such as self-attention, where the transformer can analyse every word from a sentence synchronously, therefore makes it train data effectively and can handle larger datasets, compared to RNNs which have the vanishing gradient problem. The self-attention component allows the transformer to analyse each word within a sentence and determines which word is significantly important based on the score, which highlights how relevant it is compared to the current word it is executing. This benefits the model, as it can understand the relationships of words and context in a sentence, regardless of their placement of order [49].

Encoder & Decoder:

The encoder and decoder stack, as shown in figure 4 [50] is a transformer model that was first introduced for machine translation activities. It consists of $N = 6$ similar layers within the stack [50]. The encoder starts by receiving the complete version of a sentence as input and gets split into tokens. Each token is then converted into a word embedding and is passed down to each layer of the encoder. The encoder which is made up of a Multi-Head Attention, Add & Norm and Feed Forward block [51]. This helps the encoder log contextual meaning and relationships across each location within the input, to help the model understand [51] [52].

This is then supplied to the decoder to produce the output text, based on the contextual meaning and relationships from the encoder as well as the tokens generated from the decoder. It does this one token at a time, through a step-by-step process known as autoregressive. The decoder will then predict its next token and the process will be repeated until the output is complete [51] [50].

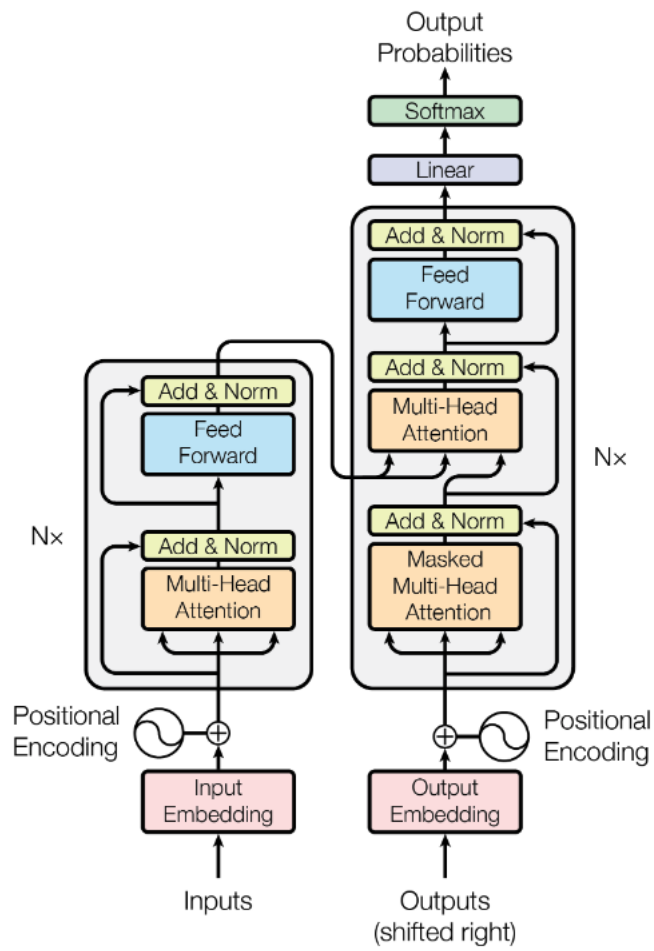


Figure 7 - A diagram representing the transformer architecture [50]

Methodology

The research methodology that will be chosen for this project will be an experimental approach. This approach will be chosen because the project aims to evaluate and compare three models to see which model is the best performing model in detecting offensive vehicle registration numbers (VRN), under a controlled condition. The experimental methodology fits the project as it will be based on an area of machine learning. Therefore, the project will then look to test the research question in looking into which transformer model will deliver the best performance on offensive VRNs.

This will begin with a thorough review of the current literature to understand what is currently known and the gap that is to be presented through the project. It will firstly start by looking into what the current approach in detecting offensive VRNs and the process within the Driver and Vehicle Licensing Agency (DVLA). Secondly, it will look at what the current technology is, using Natural Language Processing (NLP), specifically based on the offensive language domain and VRNs. Thirdly, a review of Machine Learning (ML) will conduct to review techniques and concepts that may be considered for the project. The research will then proceed to the focus of Deep Learning (DL) and to provide an overview of neural network models and dive deep into transformer models such as BERT, RoBERTa and DistilBERT.

The research will be conducted using academic papers, journals and research papers across digital libraries such as IEEE Xplore, ACM Digital Library, ArXiv, SpringerLink

and Google Scholar. Peer reviewed journal articles will be prioritised first then conference and research papers.

This process has been chosen to ensure that only high quality and peer reviewed academic sources are chosen to produce a strong and robust literature review on the topic of the project. The academic papers that will be used within the literature review will be published in the past five years to ensure the studies capture the latest recent developments of Artificial Intelligence (AI), ML, NLP, DL etc. Seminal papers may be included which might be older than the five-year publication date, which will be relevant to the area of study.

The analysis of the dataset process will investigate the raw English suppressed list obtained from the DVLA which holds all the banned offensive VRNs held internally. The analysis will firstly look to identify the type of offensive VRNs that are held within the dataset. Followed by the initial dataset, a subsequent dataset for non-offensive VRN's will follow. This will include normal personalised VRNs. The analysis is required to ensure that a coherent and structured format will follow in the preprocessing and cleaning of the dataset step.

The design activity process will begin by designing a dataset format that will include two columns. The first column will include the VRN of the offensive and non-offensive VRN and the second will be a label which will correspond to either zero for non-offensive or one for offensive. This follows the binary classification within ML to categorise the classes into separate sections. This structure will follow a supervised learning approach, due to the datasets being identified with the required categories. The process will then follow a preprocess stage within the design process. This will begin by

converting the raw datasets into data that can be used to train the model effectively. This starts by removing text such as dates, policies, headers etc. Then a removal of special characters such as asterisks that have been applied to the majority of the VRNs are to be removed from the dataset. Each VRN within the offensive dataset, will strip the whitespace to allow the model to detect patterns in a simple way.

Once both datasets have been cleaned and pre-processed. Both datasets will then be merged into one dataset to act as the training dataset for the model to train on, with data shuffled in random order to prevent the model from learning and leaning towards one side, which will prevent bias. Once this has been done, an 80/20 split will be introduced, where 80% of the dataset is left for training and 20% of data is extracted into a separate dataset, which will be effectively used for testing purposes and how well the trained model can generalize, to prevent any class imbalance, during the predications of the model.

Following on further from the design process, a tokenization approach will be applied to each model for BERT, RoBERTa and DistilBERT. The models will each use a tokenizer such as Bert-Base-Uncased [55] , Roberta-base [56] and DistilBERT-base-uncased [57]. The tokenizer is required to be applied to each dataset so it can allow the model to convert words to numbers to allow it to understand the VRN during training. The parameters the model will use are padding, truncation and a max length. This has been chosen to make all the sequences in the batch the same length and ensure

processing speed will not be slow. The label encoder will apply to the dataset to handle each label to the category of either offensive or non-offensive when it is being trained.

BERT was chosen as it was one of the strongest transformer models for detecting offensive language and hate speech. While DistilBERT and RoBERTa offered similar results, they are all state-of-the-art models as outlined in the literature review. Therefore it was justified to take all three models and run a comparison on what model will perform better.

The evaluation & criteria to be used within this study will be performance metrics. Specifically, accuracy, f1-score, precision and recall. Accuracy has been chosen to predict what percentage of the model predicts correct on offensive and non-offensive VRNs. The F1-Score is chosen as it will indicate if the predication of the model is imbalanced, which will be important as the results should be fair and split equally. The precision metric will be based on how many VRNs the model will predict correctly, and Recall will be for how many VRNs were identified correctly during the training process.

The materials that will be used for the implementation is the Hugging Face framework. This will be chosen as it is the main area to learn and develop a transformer model, which have been pretrained on a large corpus of data. Therefore, this will be the main focus of this study. Hugging Face is documented with technical and step by step code which enables this project to be developed using state of the art transformer models based on research that has been academically studied prior to this study. An example

would be the pretrained transformer model BERT-base-uncased [55] to detect offensive language, which has been built by Google researchers.

The language chosen will be Python as it an industry standard within ML and NLP activities. Python is suitable for this project as it has documented tutorials and academic papers based on Transformers and ML, which will inevitably help the research become stronger. The use of Python will also ensure that pre-trained models such as BERT, DistilBERT and RoBERTa can be used straight away, without extra configuration required, due to the imports of libraries and frameworks from the Hugging Face framework. Furthermore, the use of Pandas will be required as it will ensure that saving and loading datasets can be done with ease. As well as splitting the dataset to ensure the models' training data is equally split for train and test validation, ensures that less time is required to do these manually, which can be time consuming and prone to errors. PyTorch will be used for training the model and inference. This is specifically important as it will be required for the prediction of the model when it looks at the unseen dataset.

The evaluate library will be used, as this will handle the performance metrics outlined above and will be essential in comparing which transformer model in this study, will perform best on all metrics. The Numpy library will be used for the computational methods when training and predicting the models results. Matplotlib and Sklearn libraries will be both used for generating of the confusion matrix and classification report table. The hardware that will be used for this project will include an Apple Macbook Pro M2, with 32GB of RAM, 512GB SSD storage. The software will include the use of PyCharm as the IDE to develop the implementation, Git to track changes,

GitHub to save the repository in the cloud, Microsoft word to write up the dissertation and Zotero to keep track of references.

Results and Discussions

Table 4 presents the results for the BERT classification report on a pre-trained model based on predictions of an unseen dataset. The classification report uses two classes, which evaluate the level of how the model performed across the non-offensive and offensive classes. Within table 4, the performance metrics measured across each class are precision, recall, F1-Score, support, accuracy, macro avg and weighted avg.

	Precision	Recall	F1-Score	Support
Non-Offensive	0.86	1.0	0.92	1189.0
Offensive	1.0	0.83	0.91	1189.0
accuracy	0.92	0.92	0.92	0.92
macro avg	0.93	0.92	0.92	2378.0
weighted avg	0.93	0.92	0.92	2378.0

Table 4 - Results for BERT classification report

Table 5 presents the results for the RoBERTa classification report on a pre-trained model based on predictions of an unseen dataset.

	Precision	Recall	F1-Score	Support
Non-Offensive	0.86	1.0	0.93	1189.0
Offensive	1.0	0.84	0.91	1189.0
accuracy	0.92	0.92	0.92	0.92
macro avg	0.93	0.92	0.92	2378.0
weighted avg	0.93	0.92	0.92	2378.0

Table 5 - RoBERTa classification report

Table 6 presents the results for the DistilBERT classification report on a pre-trained model based on predictions of an unseen dataset.

	Precision	Recall	F1-Score	Support
Non-Offensive	0.87	1.0	0.93	1189.0
Offensive	1.0	0.85	0.92	1189.0
accuracy	0.93	0.93	0.93	0.93
macro avg	0.94	0.93	0.93	2378.0
weighted avg	0.94	0.93	0.93	2378.0

Table 6 - DistilBERT classification report

A confusion matrix for the BERT transformer model based on unseen data predictions as shown in figure 8. This highlights a visual output.

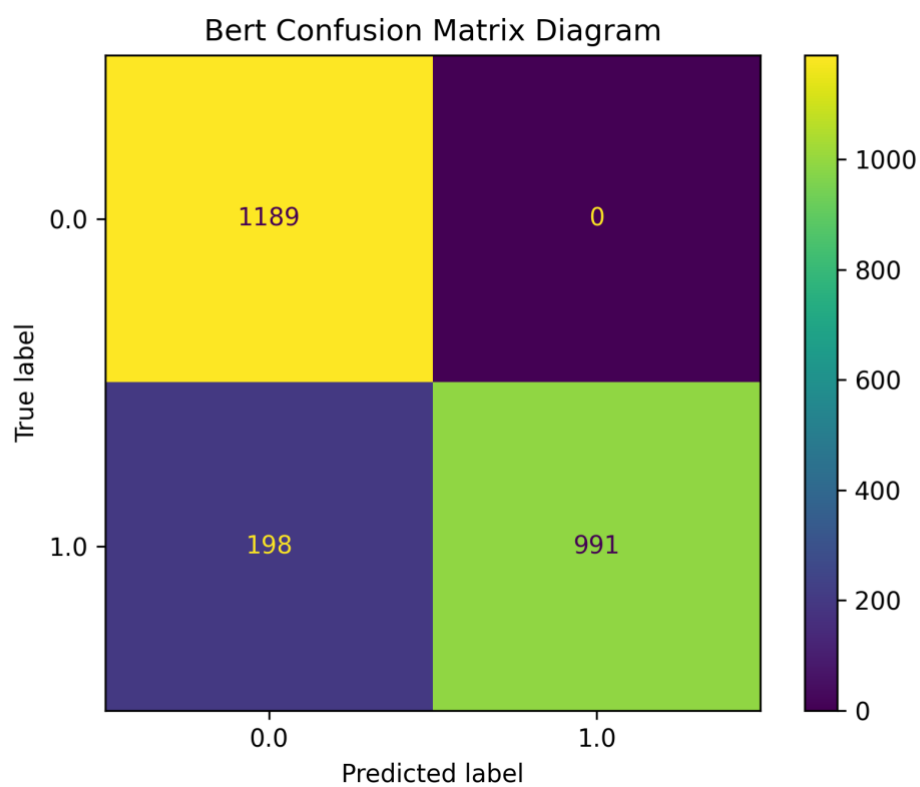


Figure 8 - BERT confusion matrix

Figure 9 presents the results for the RoBERTa model using a confusion matrix.

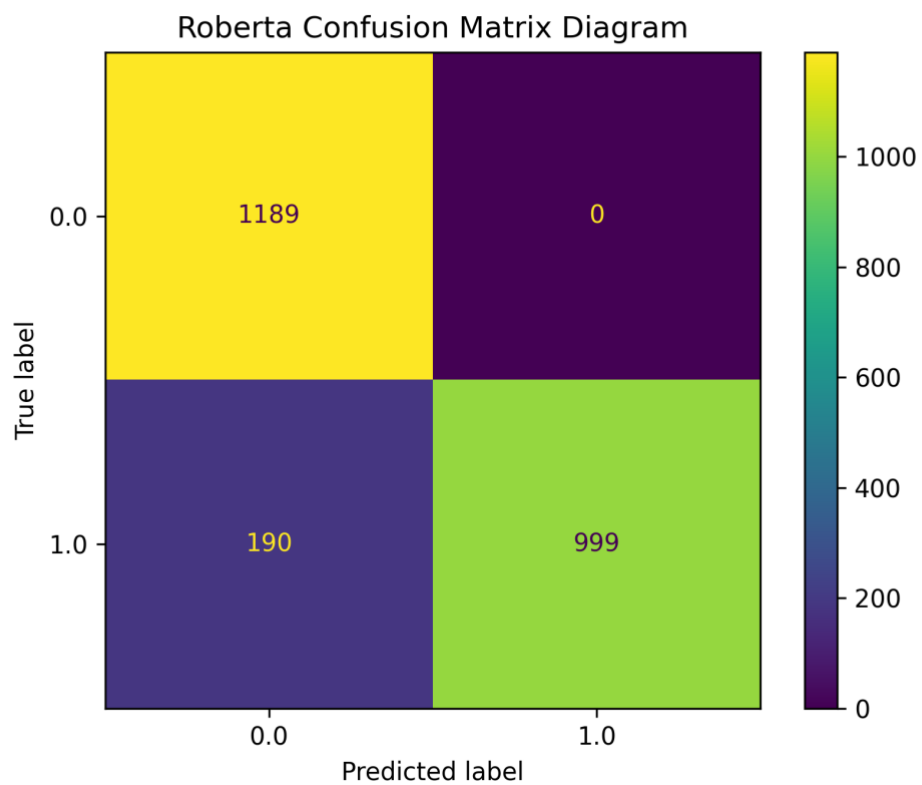


Figure 9 - RoBERTa confusion matrix

Figure 10 presents the results for the DistilBERT model using a confusion matrix.

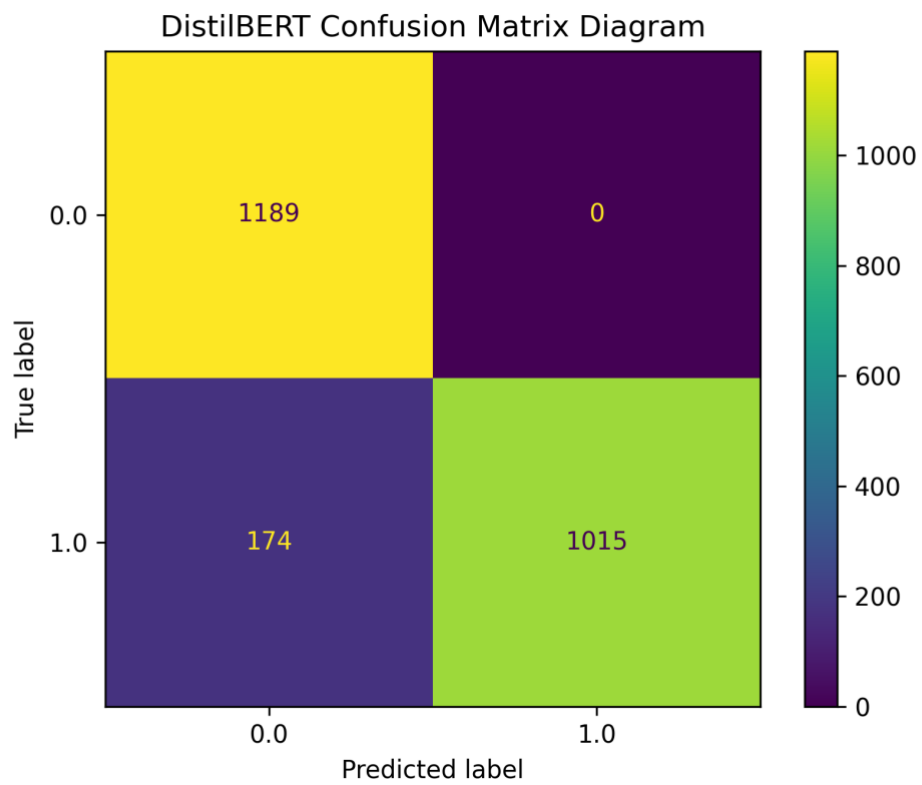


Figure 10 – DistilBERT confusion matrix

Discussion:

The BERT transformer model indicates a strong performance of 92% accuracy spanning across the entire dataset which holds a total of 2378 VRNs. This demonstrates how effective the model is with detecting VRNs based on offensive and non-offensive data on unseen data.

The accuracy is complemented well with a 100% precision rate for the offensive model. The result means that the models' offensive predictions are always correct, 100% of the time, which means that there were no false positives.

However, the recall of the model is 83% for the offensive dataset, which implies that the model can catch the majority of VRNs. That said, it has still missed 17% of VRNs. In the context of content moderation for detecting offensive VRNs, 202 out of 1189 is a high number to slip through the nets. This poses a risk for users, as VRNs can then still be purchased and be displayed on vehicles. Therefore, it is important to reduce this from happening, by improving the recall as much as possible.

While the model is good at detecting offensive VRNs. The non-offensive has the opposite effect, where the model was only able to identify 86% of the non-offensive VRN's based on the unseen data. This reveals that, 14% of the time the model predicts VRNs as non-offensive, they were indeed offensive, which means that there are false positive predictions made.

This suggests that out of the 166 VRNs, the model is not detecting properly, therefore it is missing the VRN's when predicting on unseen data and incorrectly labelling the data

as non-offensive. This could imply that the model has not trained well enough on similar examples of the VRNs during training, therefore wrongly predicting.

Out of all the non-offensive data, the model successfully captures a 100% recall rate, which reveals that it does not miss any VRNs that were classified as non-offensive. Therefore, this implies that non-offensive VRNs are safe.

A note worth pointing out, is that both classes; offensive and non-offensive, have a support of 1189 each, which indicate that the datasets are balanced and fair when running this experiment. This is important as it ensures the trained model does not lean towards one side over the other, which can decrease performance metrics.

RoBERTa achieved a similar performance compared to BERT, leading with 92% accuracy. However, it indicates a similar opposite effect with the precision being 86% but capturing 100% recall for non-offensive VRN's while maintaining a 100% precision rate, with a 84% recall rate. This demonstrates the transformer model performs incredibly well for offensive and non-offensive language detection.

DistilBERT results indicate that the model has performed a strong performance, which has outperformed both BERT and RoBERTa in the detection of offensive and non-offensive VRN's based on unseen data. A 93% accuracy, which is a 1% increase from the previous two models, show that despite the model being a smaller version, it is still able to perform well compared to bigger large language models within transformers. DistilBERT was able to achieve an increase on every metric within the classification report. This signifies that DistilBERT is the best model both in detection of offensive and non-offensive VRN's, for the precision, recall, f1-score and accuracy in terms of

this research project. By comparing the results found within this study, with previous literature, it had indicated that BERT was the most suitable transformer model and was the strongest in detecting offensive language. However, that said, with the results shown in this study, that is not the case and DistilBERT has proven to be a top contender within detection of offensive language.

As shown in figure 8, is a confusion matrix for the BERT classified model based on unseen data. The top left corner indicates that there were 1189 True Negatives (TN), where the model predicted non-offensive and it was correct. The top right corner states that zero were predicted for False Positives (FP), which indicate that the model predicted as offensive, but was actually non-offensive. The bottom left corner, states that 198 were marked as False Negatives (FN), where the model had predicted VRNs as non-offensive, but were actually offensive. Lastly is the bottom right corner, which suggests that there were 991 True Positives (TP), where the model predicted as offensive and it was correct.

A similar experience as BERT is shown for RoBERTa on the confusion matrix in figure 9. This provides a visual representation in how the model predicted offensive and non-offensive data.

In figure 10, the DistilBERT confusion matrix is shown. TP signifies an increase in offensive predications made correctly by the model, compared to the previous two models. This also highlights that the FN is much lower than the two models, which signifies that the model predicted that VRNs were non-offensive, meanwhile they were offensive instead. But that is a reduced number, which indicates good results for the best performing model in this study.

Conclusion and future work:

The project aims were to evaluate the current literature and understand the state of the field within NLP and the offensive language within vehicle registration numbers. To achieve the work, it involved looking into academic papers, journals and papers to find out what technology and processes are used to build a tool that could allow the detection of offensive number plates to be achieved. This was achieved using a combination of deep learning and machine learning to achieve the evaluation metrics that were performed at the end of the study. It focuses on the model performing on training data which included offensive and non-offensive number plates to then be tested on a unseen dataset which contains a mixture of both offensive and non-offensive data, which were shuffled and presented to the model to be able to set predictions on whether it was able to generalize well to the dataset.

However a notable issue that was identified was that prior to the work starting for the detection of offensive vehicle number plates in Arabizi, there was not a suitable dataset large enough for the experiment to run or to be created manually due to the lack of the writers knowledge of the area of offensive words to create the dataset. Therefore the project was changed and replaced to English and use DVLA official datasets which were supplied in December 2025. Allowing the study to take place and run experiments, which have proven to be a success.

Future works to take place would be to expand the transformer models and to improve the current evaluation metrics that have been demonstrated within the results and discussions of this paper.

A reflection on the entire process, has provided me with immense knowledge of the area of NLP, deep learning and machine learning. It has improved my knowledge and experience in analysing a problem within the DVLA, coming up with ideas and solutions to resolve the problem, through the use of ML and DL. It has also allowed me to develop my skills within Python and pandas to develop such solution.

Bibliography

- [1] M. F. Arif and J. Nayak, 'Fine-Tuned BERT Model for Accurate Hate Speech Detection in Social Media', in *2024 International Conference on Modeling, Simulation & Intelligent Computing (MoSICom)*, Dubai, United Arab Emirates: IEEE, Dec. 2024, pp. 406–410. doi: 10.1109/MoSICom63082.2024.10880992.
- [2] M. Chakarverti, A. Goswami, and A. Yadav, 'Comparative Evaluation of Pre-Trained Models for Hate Speech Detection on Social Media', in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kamand, India: IEEE, Jun. 2024, pp. 1–6. doi: 10.1109/ICCCNT61001.2024.10723959.
- [3] B. S. Kasab, G. D. Jadhav, S. B. Nalwade, S. Tikore, K. Vatekar, and D. Lavate, 'Real-Time Vehicle Plate Detection and Recognition', in *2025 World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS)*, Indore, India: IEEE, Aug. 2025, pp. 1–6. doi: 10.1109/WorldSUAS66815.2025.11199079.
- [4] R. Arora, M. Agarwal, V. N A S, U. Maheswari K M, and B. R, 'Vehicle License Plate Detection and Recognition using YOLOv9 and Easy OCR', in *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, Chennai, India: IEEE, Mar. 2025, pp. 1–6. doi: 10.1109/ICDSAAI65575.2025.11011682.
- [5] N. M. Kailash Varma, S. H. Mattaparty, S. Ismail, J. Thaduri, G. Deep Arora, and A. B, 'Sentiment Analysis: A Machine Learning Perspective', in *2024 First International Conference on Electronics, Communication and Signal Processing (ICECSP)*, New Delhi, India: IEEE, Aug. 2024, pp. 1–6. doi: 10.1109/ICECSP61809.2024.10698402.
- [6] A. S. Jaiswal, D. V. Bhavsagar, K. Dhole, S. Chaurasia, M. Daph, and S. Chourasia, 'Sentiment Analysis of Election Result Prediction using Twitter Data By NLP and ML', in *2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET)*, Nagpur, India: IEEE, Jun. 2024, pp. 1–4. doi: 10.1109/ICICET59348.2024.10616319.
- [7] B. Kumar, Sheetal, V. S. Badiger, and A. Ds. Jacintha, 'Sentiment Analysis for Products Review based on NLP using Lexicon-Based Approach and Roberta', in *2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, Bangalore, India: IEEE, Jan. 2024, pp. 1–6. doi: 10.1109/IITCEE59897.2024.10468039.
- [8] M. R. Kounte, P. K. Tripathy, P. P., and H. Bajpai, 'Analysis of Intelligent Machines using Deep learning and Natural Language Processing', in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, Tirunelveli, India: IEEE, Jun. 2020, pp. 956–960. doi: 10.1109/ICOEI48184.2020.9142886.
- [9] Mahyar Amini and Yaser Akbari, 'Operating Machine Learning across Natural Language Processing Techniques for Improvement of Fabricated News Model', *ResearchGate*, Oct. 2022, [Online]. Available: https://www.researchgate.net/publication/364340252_Operating_Machine_Le

arning_across_Natural_Language_Processing_Techniques_for_Improvement_of_Fabricated_News_Model

- [10] R. N. Thomas and R. Gupta, 'A Survey on Machine Learning Approaches and Its Techniques', in *2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, Bhopal, India: IEEE, Feb. 2020, pp. 1–6. doi: 10.1109/SCEECS48394.2020.190.
- [11] A. Sharma, A. Kaur, and A. Semwal, 'Supervised and Unsupervised Prediction Application of Machine Learning', in *2022 International Conference on Cyber Resilience (ICCR)*, Dubai, United Arab Emirates: IEEE, Oct. 2022, pp. 1–5. doi: 10.1109/ICCR56254.2022.9996063.
- [12] T. Jadhav et al., 'Predicting Urban Land Cover Using Classification: A Machine Learning Approach', in *2023 IEEE 11th Region 10 Humanitarian Technology Conference (R10-HTC)*, Rajkot, India: IEEE, Oct. 2023, pp. 450–454. doi: 10.1109/R10-HTC57504.2023.10461930.
- [13] V. Gupta, V. K. Mishra, P. Singhal, and A. Kumar, 'An Overview of Supervised Machine Learning Algorithm', in *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*, Moradabad, India: IEEE, Dec. 2022, pp. 87–92. doi: 10.1109/SMART55829.2022.10047618.
- [14] P. Shengguang, 'Overview of Meta-Reinforcement Learning Research', in *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, Guangzhou, China: IEEE, Dec. 2020, pp. 54–57. doi: 10.1109/ITCA52113.2020.00019.
- [15] C. Richardson, S. Shah, and X. Yuan, 'Semi-Supervised Machine Learning for Analyzing COVID-19 Related Twitter Data for Asian Hate Speech', in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, Nassau, Bahamas: IEEE, Dec. 2022, pp. 1643–1648. doi: 10.1109/ICMLA55696.2022.00252.
- [16] S. Alsafari and S. Sadaoui, 'Semi-Supervised Self-Learning for Arabic Hate Speech Detection', in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Melbourne, Australia: IEEE, Oct. 2021, pp. 863–868. doi: 10.1109/SMC52423.2021.9659134.
- [17] S. Dlamini, E. Jembere, and A. Pillay, 'Evaluation of Word and Sub-word Embeddings for isiZulu on Semantic Relatedness and Word Sense Disambiguation Tasks', in *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, Durban, South Africa: IEEE, Aug. 2020, pp. 1–6. doi: 10.1109/icABCD49160.2020.9183836.
- [18] Joel Barnard, 'Word Embeddings', What are word embeddings? Accessed: Jan. 13, 2026. [Online]. Available: <https://www.ibm.com/think/topics/word-embeddings>
- [19] S. J. Johnson, M. R. Murty, and I. Navakanth, 'A detailed review on word embedding techniques with emphasis on word2vec', *Multimed. Tools Appl.*, vol. 83, no. 13, pp. 37979–38007, Oct. 2023, doi: 10.1007/s11042-023-17007-z.
- [20] D. Yang, 'Exploring Word2Vec and LSA Models for Fund Review Analysis', in *Proceedings of the 5th International Conference on Artificial Intelligence and Computer Engineering*, Wuhu Shandong China: ACM, Nov. 2024, pp. 516–521. doi: 10.1145/3716895.3716987.

- [21] I. R. Hendrawan, E. Utami, and A. D. Hartanto, 'Comparison of Word2vec and Doc2vec Methods for Text Classification of Product Reviews', in *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia: IEEE, Dec. 2022, pp. 530–534. doi: 10.1109/ICITISEE57756.2022.10057702.
- [22] G. A. McCully, J. D. Hastings, S. Xu, and A. Fortier, 'Bi-Directional Transformers vs. word2vec: Discovering Vulnerabilities in Lifted Compiled Code', in *2024 Cyber Awareness and Research Symposium (CARS)*, Grand Forks, ND, USA: IEEE, Oct. 2024, pp. 1–8. doi: 10.1109/CARS61786.2024.10778724.
- [23] S. Al-Saqqa and A. Awajan, 'The Use of Word2vec Model in Sentiment Analysis: A Survey', in *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control*, Cairo Egypt: ACM, Dec. 2019, pp. 39–43. doi: 10.1145/3388218.3388229.
- [24] B. Kumar, C. Srivastava, and A. Tripathi, 'Sarcasm Detection in Social Media Text Using GloVe Word Embeddings in Machine Learning', in *2025 3rd International Conference on Communication, Security, and Artificial Intelligence (ICCSAI)*, Greater Noida, India: IEEE, Apr. 2025, pp. 1–6. doi: 10.1109/ICCSAI64074.2025.11064104.
- [25] Jeffery Pennington, Richard Socher, and D Christopher Manning, 'Glove: Global vectors for word representation', presented at the Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2014, pp. 1532–1543. Accessed: Jan. 15, 2026. [Online]. Available: <https://aclanthology.org/D14-1162.pdf>
- [26] A. S. Prapty and K. M. A. Hasan, 'Semantic Clustering of Bangla Natural Word using Different Word Embedding Techniques', in *2022 25th International Conference on Computer and Information Technology (ICCIT)*, Cox's Bazar, Bangladesh: IEEE, Dec. 2022, pp. 1052–1056. doi: 10.1109/ICCIT57492.2022.10054703.
- [27] W. R. Arianto et al., 'Comparative Study of Word2Vec, FastText, and Glove Embeddings for Synonym Identification in Bugis Language', in *2024 Beyond Technology Summit on Informatics International Conference (BTS-I2C)*, Jember, East Java, Indonesia: IEEE, Dec. 2024, pp. 555–560. doi: 10.1109/BTS-I2C63534.2024.10942212.
- [28] R. K. Giri, S. C. Gupta, and U. K. Gupta, 'An approach to detect offence in Memes using Natural Language Processing(NLP) and Deep learning', in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India: IEEE, Jan. 2021, pp. 1–5. doi: 10.1109/ICCCI50826.2021.9402406.
- [29] A. G. D'Sa, I. Illina, and D. Fohr, 'BERT and fastText Embeddings for Automatic Detection of Toxic Speech', in *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, Tunis, Tunisia: IEEE, Feb. 2020, pp. 1–5. doi: 10.1109/OCTA49274.2020.9151853.
- [30] C. P. Soto, G. M. S. Nunes, J. G. R. C. Gomes, and N. Nedjah, 'Application-specific word embeddings for hate and offensive language detection', *Multimed. Tools Appl.*, vol. 81, no. 19, pp. 27111–27136, Aug. 2022, doi: 10.1007/s11042-021-11880-2.

- [31] J. Kelley and M. Hagan, 'Adaptive Multistep Prediction With Sequence-to-Sequence (Seq2Seq) Models', *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 8, pp. 15561–15568, Aug. 2025, doi: 10.1109/TNNLS.2025.3525618.
- [32] R. M. Schmidt, 'Recurrent Neural Networks (RNNs): A gentle Introduction and Overview', 2019, *arXiv*. doi: 10.48550/ARXIV.1912.05911.
- [33] M. Kaur and A. Mohta, 'A Review of Deep Learning with Recurrent Neural Network', in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India: IEEE, Nov. 2019, pp. 460–465. doi: 10.1109/ICSSIT46314.2019.8987837.
- [34] S. R. Kadam and M. P. Dhore, 'Review of Deep Learning Methods', in *2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBE)*, Pune, India: IEEE, Aug. 2024, pp. 1–8. doi: 10.1109/ICCUBE61740.2024.10774807.
- [35] Z. Jing et al., 'Electricity Price Default Detection Model Based on Long Short-Term Memory Network', in *2024 IEEE 7th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, Shenyang, China: IEEE, Dec. 2024, pp. 593–597. doi: 10.1109/AUTEEE62881.2024.10869763.
- [36] M. Bkassiny, 'A Deep Learning-based Signal Classification Approach for Spectrum Sensing using Long Short-Term Memory (LSTM) Networks', in *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia: IEEE, Dec. 2022, pp. 667–672. doi: 10.1109/ICITISEE57756.2022.10057728.
- [37] N. Raghunathan, P. Shiwakoti, M. P. A. M., and S. Shanmugam, 'Sentiment Analysis for Product Reviews using Long Short-Term Memory with Layer Normalization', in *2023 First International Conference on the Advancements of Artificial Intelligence in African Context (AAIAC)*, Arusha, Tanzania, United Republic of: IEEE, Nov. 2023, pp. 1–6. doi: 10.1109/AAIAC60008.2023.10465282.
- [38] S. Das and D. Das, 'Natural Language Processing (NLP) Techniques: Usability in Human-Computer Interactions', in *2024 6th International Conference on Natural Language Processing (ICNLP)*, 2024, pp. 783–787. doi: 10.1109/ICNLP60986.2024.10692776.
- [39] J. Francis and M. Subha, 'An Overview of Natural Language Processing (NLP) in Healthcare: Implications for English Language Teaching', in *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Kirtipur, Nepal: IEEE, Oct. 2024, pp. 824–827. doi: 10.1109/I-SMAC61858.2024.10714890.
- [40] Amazon, 'Alexa', Alexa. [Online]. Available: <https://alexa.amazon.co.uk/>
- [41] Apple, 'Siri', Siri. [Online]. Available: <https://www.apple.com/siri/>
- [42] Google, 'Google Assistant', Google Assistant. [Online]. Available: <https://assistant.google.com/>
- [43] S. Ravichandiran, *Getting started with Google BERT: build and train state-of-the-art natural language processing models using BERT*. Birmingham Mumbai: Packt, 2021.
- [44] N. Mossad, Y. Mohamed, A. Fares, and A. B. Zaky, 'Arabic text sentiment analysis and emotion classification using transformers', in *2023 11th International Japan-Africa Conference on Electronics, Communications, and*

- Computations (JAC-ECC)*, Alexandria, Egypt: IEEE, Dec. 2023, pp. 131–137. doi: 10.1109/JAC-ECC61002.2023.10479609.
- [45] Y. Liu, H. Huang, J. Gao, and S. Gai, ‘A study of Chinese Text Classification based on a new type of BERT pre-training’, in *2023 5th International Conference on Natural Language Processing (ICNLP)*, Guangzhou, China: IEEE, Mar. 2023, pp. 303–307. doi: 10.1109/ICNLP58431.2023.00062.
- [46] A. Pathak and S. Rathor, ‘Abusive Content Detection with LangChain Integration Using Fine-Tuned RoBERTa’, in *2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT)*, Bhopal, India: IEEE, Mar. 2025, pp. 1–5. doi: 10.1109/CSNT64827.2025.10968367.
- [47] G. Brauwers and F. Frasincar, ‘A General Survey on Attention Mechanisms in Deep Learning’, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3279–3298, Apr. 2023, doi: 10.1109/TKDE.2021.3126456.
- [48] N. Zhang and J. Kim, ‘A Survey on Attention mechanism in NLP’, in *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, Singapore: IEEE, Feb. 2023, pp. 1–4. doi: 10.1109/ICEIC57457.2023.10049971.
- [49] D.-H. Hoang, A.-K. Tran, D. N. Minh Dang, P.-N. Tran, H. Dang-Ngoc, and C. T. Nguyen, ‘RBBA: ResNet - BERT - Bahdanau Attention for Image Caption Generator’, in *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea, Republic of: IEEE, Oct. 2023, pp. 430–435. doi: 10.1109/ICTC58733.2023.10392496.
- [50] A. Vaswani et al., ‘Attention Is All You Need’, Aug. 02, 2023, *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.
- [51] S. Kim et al., ‘Full Stack Optimization of Transformer Inference: a Survey’, Feb. 27, 2023, *arXiv*: arXiv:2302.14017. doi: 10.48550/arXiv.2302.14017.
- [52] X. Tan, Y. Lei, X. Tan, L. Yang, K. Huang, and S. He, ‘Research on Emotional Transfer in Classical Chinese Poetry Using a Multi-Module Transformer-Based Deep Learning Framework’, in *Proceedings of the 2nd International Conference on Intelligent Computing and Data Analysis*, Zhengzhou China: ACM, Aug. 2025, pp. 884–893. doi: 10.1145/3772726.3772865.
- [53] K. Vayadande, A. M. Kulkarni, G. B. Yadav, R. Kumar, and A. R. Sawant, ‘A Comprehensive Analysis of Various Tokenization Techniques and Sequence-to-Sequence Model in Natural Language Processing’, in *How Machine Learning is Innovating Today’s World*, 1st edn, A. Dey, S. Nayak, R. Kumar, and S. N. Mohanty, Eds, Wiley, 2024, pp. 1–11. doi: 10.1002/9781394214167.ch1.
- [54] R. Omar and H. Muccini, ‘Greening the AI: Evaluating tokenization methods for Energy-efficient NLP’, in *2025 11th International Conference on ICT for Sustainability (ICT4S)*, Dublin, Ireland: IEEE, Jun. 2025, pp. 133–142. doi: 10.1109/ICT4S68164.2025.00022.
- [55] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’, *CoRR*, vol. abs/1810.04805, 2018, [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [56] Y. Liu et al., ‘RoBERTa: A Robustly Optimized BERT Pretraining Approach’, *CoRR*, vol. abs/1907.11692, 2019, Accessed: Jan. 31, 2026. [Online]. Available: <http://arxiv.org/abs/1907.11692>

- [57] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, 'DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter', *ArXiv*, vol. abs/1910.01108, 2019.
- [58] M. Saunders, P. Lewis, and A. Thornhill, *Research methods for business students*, 4. ed. Harlow: Financial Times Prentice Hall, 2007.
- [59] Driver and Vehicle Licensing Agency and Abdulmalik Elghedafi, 'DVLA Datasets', DVLA Atlassian. Accessed: Feb. 01, 2026. [Online]. Available: <https://dvla.atlassian.net/wiki/x/5Q0Nqg>
- [60] GPT4, 'AutoTokenizer', James McGuigan. Accessed: Jan. 31, 2026. [Online]. Available: https://www.jamesmcguigan.com/content/llm/generative_ai/huggingface/transformers/AutoTokenizer.html
- [61] HuggingFace, 'Loading a Dataset', Loading a Dataset. Accessed: Jan. 31, 2026. [Online]. Available: https://huggingface.co/docs/datasets/v1.2.1/loading_datasets.html
- [62] 'Transformers for Text Classification with IMDb Reviews', PyTorch-Ignite. Accessed: Jan. 31, 2026. [Online]. Available: <https://pytorch-ignite.ai/tutorials/beginner/02-transformers-text-classification/>
- [63] Hugging Face, 'Text classification', HuggingFace. Accessed: Jan. 31, 2026. [Online]. Available: https://huggingface.co/docs/transformers/tasks/sequence_classification
- [64] HuggingFace, 'Metrics', HuggingFace. Accessed: Jan. 31, 2026. [Online]. Available: https://huggingface.co/docs/datasets/main/en/how_to_metrics
- [65] 'Evaluate Metric Accuracy', Hugging Face. Accessed: Dec. 11, 2025. [Online]. Available: <https://huggingface.co/spaces/evaluate-metric/accuracy>
- [66] 'Evaluate-Metric F1', Hugging Face. Accessed: Dec. 11, 2025. [Online]. Available: <https://huggingface.co/spaces/evaluate-metric/f1>
- [67] 'Evaluate-Metric precision', Hugging Face. Accessed: Dec. 11, 2025. [Online]. Available: <https://huggingface.co/spaces/evaluate-metric/precision>
- [68] 'Evaluate-Metric recall', Hugging Face. Accessed: Dec. 11, 2025. [Online]. Available: <https://huggingface.co/spaces/evaluate-metric/recall>
- [69] Hugging Face, 'Trainer', Hugging Face. Accessed: Jan. 31, 2026. [Online]. Available: https://huggingface.co/docs/transformers/main_classes/trainer
- [70] Hugging Face, 'Transformer Notebooks', Hugging Face. Accessed: Jan. 31, 2026. [Online]. Available: <https://huggingface.co/docs/transformers/notebooks>
- [71] Facebook AI and HuggingFace, 'RoBERTa', Hugging Face. Accessed: Jan. 31, 2026. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/roberta
- [72] 'Pipelines', Hugging Face. Accessed: Jan. 31, 2026. [Online]. Available: https://huggingface.co/docs/transformers/main_classes/pipelines
- [73] EdChum, 'Python: Convert dataframe into a list with string items inside list', Stack overflow. Accessed: Jan. 31, 2026. [Online]. Available: <https://stackoverflow.com/questions/41768196/python-convert-dataframe-into-a-list-with-string-items-inside-list>
- [74] Nicoberk, 'How to set useful id2label and label2id in config.json using Trainer', Hugging Face. Accessed: Jan. 31, 2026. [Online]. Available: <https://discuss.huggingface.co/t/how-to-set-useful-id2label-and-label2id-in-config-json-using-trainer/43545>

- [75] 'Behind the pipeline', Hugging Face. Accessed: Jan. 31, 2026. [Online]. Available: <https://huggingface.co/learn/llm-course/chapter2/2#preprocessing-with-a-tokenizer>
- [76] Nikolas Rieble and desertnaut, 'Getting the accuracy from classification_report back into a list', Stack overflow. Accessed: Jan. 31, 2026. [Online]. Available: <https://stackoverflow.com/questions/56870373/getting-the-accuracy-from-classification-report-back-into-a-list>
- [77] scikit-learn developers, 'classification_report', Scikit-learn. Accessed: Jan. 31, 2026. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html
- [78] janus235 and desertnaut, 'scikit learn output metrics.classification_report into CSV/tab-delimited format', Stack overflow. Accessed: Jan. 31, 2026. [Online]. Available: <https://stackoverflow.com/a/53780589>

Appendix:

The below in-text citations are used for adding the references from the implementation to the bibliography.

English suppressed dataset provided by the DVLA from my previous line manager on MS Teams November 2025:

Offensive dataset (English suppressed list of VRNs): [59]

Non offensive dataset (personalised registration VRNs): [59]

Auto Tokenizer: [60]

Loading dataset: [61]

Tokenize function: [62]

Sequence classification: [63]

Metrics: [64] [65] [66] [67] [68]

Trainer: [69]

Transformers examples: [70]

RoBERTa: [71] [56]

Pipelines: [72]

Data Frame: [73]

Label Encoding: [74]

Saving finetuned model: [74]

Prediction mode: [75]

Classification report: [76] [77] [78]

A table which includes the abbreviation and full word of what it means within the project:

Abbreviation	Full word
VRN	Vehicle Registration Number
DVLA	Driver and Vehicle Licensing Agency
NLP	Natural Language Processing
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
BERT	Bidirectional Encoder Representations from Transformers
RoBERTa	Robustly Optimized BERT Pretraining Approach
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
F1	

Project Proposal:

Date	Start time	End time	Activity Description
01/08/2025	20:00	00:00	Made notes for project proposal, listening back to the recording that Tim discussed in April to understand what I need to do.
02/08/2025	13:00		Drafted logbook created repository on GitHub for project and populating the project template in latex. Read project handbook. Started

			looking into my project idea
2nd October	4pm	6pm	Submitted my MSc project proposal on to Moodle
3 rd October	12pm	12:30pm	Going to be having my meeting with Mark today.
5 th October	12:00pm	10pm	Creating the dataset, researching inappropriate words for Arabic and converting to Arabizi for the project and creating different variations of each word.
11 th October	Afternoon	Late evening	Researched academic papers and made notes on aims and directions, limitations and gaps of authors papers.
14 th to 23 rd October	All day	All day	Travelling, on Holiday. I could not manage to work on the project.
24 th October	4pm	5pm	Had project meeting with Mark, he mentioned that my grade would be around the 70% mark. He has provided me with constructive criticism.
29 th October	8pm	10pm	Started on writing the section on Techniques within NLP such as Tokenization.
30 th October	7pm	11pm	Researching more into Tokenization within NLP and

			wrote about two methods.
31 st October			Completed the Tokenization within NLP part. Had a project meeting with Mark, discussed the bad word list at DVLA and how I could use that for the project i.e development and literature.
1 st November	6pm	11pm	Researching about part of speech tagging using a rule-based approach and doing critical analysis on this section
2 nd November	4pm		Completed POS tagging section and critical analysis.
3 rd to 7 th November	N/A	N/A	Week off, as was drained mentally from work on a Springboot course. So didn't have the capability to work on the MSc project.
9 th November	3:30pm		Working on Sentiment Analysis for literature review.
17 th to 21 st November	Morning	Evening	Completed Machine learning and critical analysis
21 st November			Had Meeting with Mark and discussed what I did during the week. Then was in talks about

			changing to English instead of Arabizi for the project
22 nd November			Not much
23 rd November	Late afternoon	Evening	Looked into Hugging Face course to get started with the implementation
1 st December to 7 th December	All week		Preprocessing the datasets for offensive and non offensive using pandas.
8 th December – 10 th December			Started the implementation, tokenizing and loading dataset, training the model on the 80% of trained data.
11 th to 12 th	Morning	Night	Finished implementation for Bert model on training data and evaluating on unseen data. Also had meeting with Mark, to showcase what I did and next stages for comparison of experiment i.e introducing a second model.
13 th December			Break – not much.
14 th December	Afternoon	Night	Completed implementation on second model, And looking at analysing results.
W/C 15 th December	7pm	11pm	Started researching more information within hate speech for

			the literature review.
W/C 22nd December	7pm	11pm	Working on the literature
W/C 29th			Working on the literature
W/C 5 th January			Reading and writing about deep learning and transformers
W/C 12 th January			Continuing with Transformers section and working on the results and discussions section
W/C 19 th January	7pm	11pm	Started the results and discussions section.
W/C 26 th January	All day	All day	Completed the results and discussions, introduction, conclusion, research methodology, added more information within the literature about the transformer models i.e RoBERTa.

Project proposal:



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Detecting inappropriate license plates written in Arabizi by comparing Natural Language Processing models.

By
Abdulmalik Elghedafi

Supervisor: Mark Huntly

Project submitted as part of the requirements for
the award of
MSc Software Engineering (Web Development)

January 2026

Table of Contents

Introduction.....	60
Hypothesis	61
Project Aim.....	61
Objectives:.....	62
Literature review.....	64

Arabizi	64
Natural Language Processing (NLP)	65
BERT	65
ARABERT	66
Offensive language in Arabic	66
Preprocessing and normalisation techniques	69
Dataset	72
Toolkit	73
Motivation for this study	73
Methodology	73
Gantt Chart	76
Project planning	79
Development methods	79
Dissertation	79
Ethical	79
Expected outcomes / deliverables	80
Deliverables	80
Resources	81
Conclusion	82
Bibliography	83

Introduction

This research examines the detection of inappropriate license plates written in Latin characters and numbers known as Arabizi. A challenge of Arabizi is that it can be difficult to standardise words, due to the word having multiple variations that it can be written in, which can also cause ambiguity depending on the context. This is important as currently the DVLA have systems and rules in place to detect inappropriate number plates in English, however this project aims to fill the gap in looking at producing a

working prototype in identifying Arabizi license plates, which can be understood by Arabic speakers. The aim of the project is to compare a couple of Natural Language Processing (NLP) models that will be trained on a given dataset that includes a mix of inappropriate and non-offensive license plates and draw quantifiable data from the models, such as accuracy and performance etc.

Hypothesis

To determine whether DarijaBERT-arabizi, a Natural Language Processing (NLP) model, trained on a Arabizi dataset, will outperform AraBERT in detecting inappropriate Arabizi text based on DVLA license plates using quantifiable data.

Project Aim

The aim of the project is to evaluate and compare two language models in detecting inappropriate DVLA license plates written in Latin characters and numbers using Arabizi. The project will involve software in running the dataset and performing the output of quantifiable data

Objectives:

1. To design and develop a dataset that contains a minimum of 100 Arabizi DVLA license plates using offensive and non-offensive text.

The objective will focus on creating a new dataset that contains offensive and non-offensive text. The text will use Arabizi and be formatted in a DVLA license plate style. A minimum of 100 words will be created. Each entry of data will have different variations such as spacing and written in reverse order to stimulate against being bypassed. The dataset will undergo noise removal to clean and ensure it matches to the correct definition in Arabic. The data will be labelled as '1' for offensive and '0' for non-offensive, which marks it as identifiable when developing the software. This objective should be completed within the first week of starting the implementation of the project.

2. Implement and fine tune DarijaBERT and AraBERT models on an Arabizi dataset.

This objective will focus on using the dataset that contains offensive and non-offensive data to run each model on the same dataset while fine-tuning it to ensure it learns new data. This objective should be completed within first month of development.

3. Evaluate how well DarijaBERT NLP model did comparison to AraBERT based on quantifiable indicators to determine which model is best suited at detecting offensive text.

This objective looks at evaluating DarijaBERT and AraBERT on a custom dataset, using measurable data such as accuracy, precision time, recall and inference time. This objective will test the hypothesis. This objective should be completed within first month of development.

4. Illustrate the metrics of how well DarijaBERT and AraBERT did, by producing four visually engaging graphs by the end of the development.

The objective focuses on presenting the results and findings of the experiment, by creating four visually engaging graphs that include quantifiable metrics such as accuracy, precision, recall and inference time for both DarijaBERT and AraBERT, it allows the reader to understand which model performed the best to either prove or disprove the hypothesis.

Literature review

Arabizi

Arabizi is a form of writing text in Arabic informally through the transliteration of Latin characters and numbers in English to form words that can be both written and understood by Arabic speakers [1] . This linguistic style reflects using an English keyboard, to write Arabic text in Arabizi [2], where individuals may lack access to an Arabic keyboard. Arabizi is often used in online communications such as texting and social media [2]. People typically opt for Roman characters due to convenience and accessibility [3]. It is worth highlighting, that Arabizi does not follow standard rules and is generally avoided in formal settings [3].

In table 1, an example is shown where Arabizi is written using Latin characters and English numbers to represent the word in Arabic. Arabizi uses the number ‘7’ for ‘ح’ and number ‘9’ for ‘ق’ to make the same word in Arabic.

Arabic word	Arabizi word	Arabic letter	Number used
باعت	Ba3t	ع	3
مسألة	Ms2ala	ء	2
ختي	5ti	خ	5
حشومة	7chouma	ح	7
مقبول	Ma9boul	ق	9

Table 7- Arabizi being used with Latin characters and numbers to make a translation of a word in Arabic [3] .

An advantage of Arabizi is that communication is simplified between users that might not have access to an Arabic keyboard or are not fluent in the language. However, Arabizi can be ambiguous to understand, who are not familiar with the style of language. It is important to note that Arabizi can have multiple ways of writing a word, based on context, which can lead to uncertainty [3] [4]. It is essential in understanding Arabizi for the MSc Project, due to it being the main concept of detecting inappropriate number plates in Arabic slang written in Latin characters and numbers.

Natural Language Processing (NLP)

Natural Language Processing (NLP) is a sophisticated discipline where artificial intelligence (AI), software engineering and semantics are combined into one. NLP explores how the human language and computers can work together to listen, process and understand text. NLP can then generate text in a way that mimics the human language naturally. This is important as it allows people to communicate in a way that is normal, without requiring to learn or remember special commands for the computer to understand [5] [6]. An example of NLP are virtual assistances and chatbots, such as Alexa by Amazon [7], Siri by Apple [8], and Google assistant [9]. These are widely powered virtual assistants powered by AI, which consumers interact with everyday [5]. This is crucial for the project as it provides the methods and techniques used for text processing and interpreting Arabizi.

BERT

The **B**idirectional **E**ncoder **R**epresentation from **T**ransformer known as BERT is a potent language model developed by Google. BERT relies on the transformer architecture and specifically uses the Encoder part [10]. BERT scans text in both directions within a sentence and focuses on the context of each word in relation to

adjacent words, to better understand its meaning [11]. BERT is trained based on two stages. The first stage involves pre-training the model based on large datasets such as Wikipedia and BookCorpus [11] [10] [12]. Then it uses fine-tuning to be used on particular tasks [11].

ARABERT

AraBERT is a pre-trained language model developed to be used on Arabic. It was based on the original BERT architecture model, which has proven to show exceptional success across a wide range of NLP tasks such as detection of sarcastic tweets in Arabic and tackling false information about COVID-19 on the Twitter platform [13]. AraBERT is relevant to the project of research as it potentially may offer improved detection for Arabizi in identifying numberplates that are inappropriate in Arabic over general models such as BERT.

Offensive language in Arabic

Given the rise of social media and having unrestricted access, where users can freely share their thoughts and emotions, it can also introduce the use of offensive and hate language being sent online. In this study, the author states that while social media communities that speak in Spanish, French and English have controlled systems in place to monitor offensive language. This is not the case, in contrast for Arab countries, that speak Arabic, as tools and moderation systems are lacking to detect hateful content. In this study, they use pre-trained models to filter Arabic text for hateful and offensive detection and compares models efficiency such as BERT on whether it is offensive or neutral [14].

A strength outlined in this study shows that among all the models tested, BERT was the top performing model, which achieved superior results across all performance metrics as shown in table 2. On the other hand, AraBERT had performed strong results, due to its focus on being trained on the Arabic language [14]. When both precision and recall are well-aligned with each other, it indicates the model can detect offensive and neutral text without favouring one model over the other by false flagging offensive comments. A highlight shows that by utilising larger datasets, it can lead to the model performing exceptionally well [14].

Algorithm	Accuracy	Precision	Recall	F1-Score
BERT EN	0.98	0.98	0.98	0.98
AraBERT	0.96	0.95	0.96	0.95
LinearSVC	0.9	0.89	0.76	0.81
mBERT AR	0.83	0.84	0.82	0.83
LSTM	0.82	0.92	0.74	0.82
mBERT EN	0.81	0.82	0.80	0.81

Table 8 - language model performance scores [14]

Despite its strengths, AraBERT had incorrectly classified sarcastic comments as offensive meanwhile BERT had identified offensive words correctly. The difference stems from AraBERT being trained on a smaller dataset, which lacked exposure to pre-training on sarcastic text. However, due to BERTs, bidirectional transformer architecture. It can offer pre-training on a wider and diverse dataset and read text in both directions, to identify the context of a sentence and predict whether the text is offensive [14].

In contrast, Alshahrani and Aksoy adopt MARBERT which is a powerful language model, that has been pre-trained on a large dataset, using masking techniques [15]. The dataset was pretrained on one billion tweets in Arabic [16]. An important contribution to this study shows that MARBERT was able to achieve 85% in detecting hate speech and 93% in relation to offensive text. MARBERT had achieved accurate precision scores across a range of tested tasks. This approach worked better over the model ARABERT, as MARBERTv2 had been trained on one billion tweets in Arabic, compared to ARABERTv2-Twitter being only 60 million tweets. MARBERT can process longer sentences in Arabic text, allowing it to better understand the context and determine whether the text is offensive or hate speech [16]. This is important as words in Arabic can have different meanings depending on the context used, which can influence whether the text is non-offensive or offensive. The dataset used for MARBERT was trained on Twitter data, which contains everything from slang, to hate speech, non-offensive and offensive text.

However, the limitations of MARBERT are the lack of testing made on adversarial attacks. An example of adversarial is when a word is changed just enough to trick the model but would still convey the same input. This could be an extra space, writing the word in reverse or even using an emoji to separate the text. These changes, can confuse the model and provide incorrect predictions, but still be readable to a human [16]. This was due to the model being tested against basic tests, however the use of advanced tests, could bypass the model and provide incorrect results. It is worth noting that MARBERT was trained on Arabic dialects on Twitter, this can therefore perform lower results on other platforms and websites [16].

Despite these contributions, the authors did not address offensive text using Arabizi.

The project aims to investigate comparing DarijaBERT and AraBERT models to draw quantifiable metrics to determine which NLP model is best suited to detect offensive license plates.

Preprocessing and normalisation techniques

Due to Arabizi becoming a popular way of communicating Arabic through social media. It can present challenges due to the nature of not having an official spelling system in place. This can be increasingly difficult when working with Arabic dialects such as Moroccan Darija (MD), due to it using a mix of Spanish, English or Arabic, which can make it harder for NLP models to understand [17].

This presents an important step for the Natural Language Processing (NLP) models to follow by firstly cleaning the data through removing emojis, typos, duplicated letters before it can be used. A challenge the author highlights [17] is that inconsistent spelling can significantly reduce the models performance. Additionally, a word in Arabizi can have multiple ways to be written, therefore adding a layer of complexity to understanding and processing the text by the NLP model.

In this study, the author highlights that they used three different word lists for converting Arabizi to Arabic. The initial dataset used is Goud.ma, which is a news website for Morocco and includes 158k written articles in MD and Modern Standard Arabic using 28 million tokens. Secondly, MNAD is another dataset, that has 418k articles consisting of 96m words. This is relevant as testing and using different datasets can evaluate how well the NLP model did. The author then cleans the data by dividing the wordlist into Arabic words with a definition and misspelled words taken out. This

illustrates the claim that the dataset MD achieved a high accuracy rate of 88.4%, meanwhile the Mean Reciprocal Rank (MRR) was at 92.14% as shown in in table 3, therefore MD is the best performing method [17].

Word List	MRR	Accuracy
MD	92.14%	88.44%
Goud	88.55%	82.92%
MNAD	78.87%	71.45%

Table 9 - A table displaying the performance comparison for three datasets [17]

The author notes several drawbacks in this study, notably there is no awareness of context, when Arabizi is converted to Arabic, it can produce more than one correct word in Arabic which can lead to ambiguity. Let's take the word 'tbe3', which can be transliterated to تبع (follow) or طبع (print). These two words are often applied in Moroccan Darija. Due to not having contextual awareness, the model has a limited understanding and may choose the incorrect word, therefore reducing the model's overall performance and accuracy [17].

A further weakness identified by the author is that inconsistent spelling written in Arabizi using Latin characters can be difficult for the model to understand. This is primary because words can have multiple ways to be written, yet they can still be missed, due to not matching with what the model has, therefore this can lower the accuracy or miss words [17]. A consequence this could lead to is that words, that might have an offensive meaning, can easily be bypassed by using duplicate vowels to prolong the word, but carry the same definition and will not be recognised by the model due to the spelling.

Both studies highlight the importance of using a detailed dataset containing words to transliterate from Arabizi to Arabic. However, the author states they had used a dataset which contained 730,607 corpus and tested on 194,829 corpus for the Tunisian dialect, reaching a word error rate (WER) of 9.80% as shown in table 4 [18]. The authors findings indicate that they matched Arabizi with the equivalent word in Arabic to match their definitions using a rule-based system. Furthermore, the author included the surrounding text within a sentence as well as each letter in the word before being converted. This helps ensure the model accurately converts the Arabizi input, meanwhile interpreting the meaning of the text too [18].

	Training Corpus	Test Corpus	WER
Our Work	730,607	194,829	9.80%
Karmani et al.	500	500	17.2%
Younes et al.	36,504	9,125	4%

Table 10 - A table containing the results for the Tunisian Dialect Transliteration [18]

In comparison, it is shown that the Moroccan Arabizi paper had used three datasets, with MNAD being the largest, containing up to a staggering amount of 96 million words, making it the largest dataset compared to the study, highlighted in table 3 [17].

The paper further discusses that the Darija Open Dataset (DOD) had been chosen for the Arabizi to Arabic translation and included 10,000 words to be tested. The author had used a weighted Levenshtein method to convert the Arabizi to the correct Arabic definition and was presented with 3079 Arabizi words using MD. This led to a score of 88.4% in terms of accuracy and 92.14% in relation to the Mean Reciprocal Rank

(MRR). It is worth noting that the MD results had exceeded the MNAD and Goud reference lists in this study as shown in table 3 [17].

The weighted Levenshtein method is where the input of a word is calculated based on how many times each letter is changed or removed to match the correct spelling of the word. An example is the word ‘dawla’ in English and (دولة) in Arabic and comparing it with ‘Khawla’ (خولة) and ‘state’ (دولة). In figure 4, this is demonstrated by changing the letter from ‘ذ’ to ‘د’ and applying a smaller penalty point, based on how similar the letter sounds in Arabic which is 0.5 in this case meanwhile providing the correct transliteration. If the model had provided the wrong transliteration, this would be awarded a penalty of one point, as the letter ‘خ’ is not the same sounding letter [17].

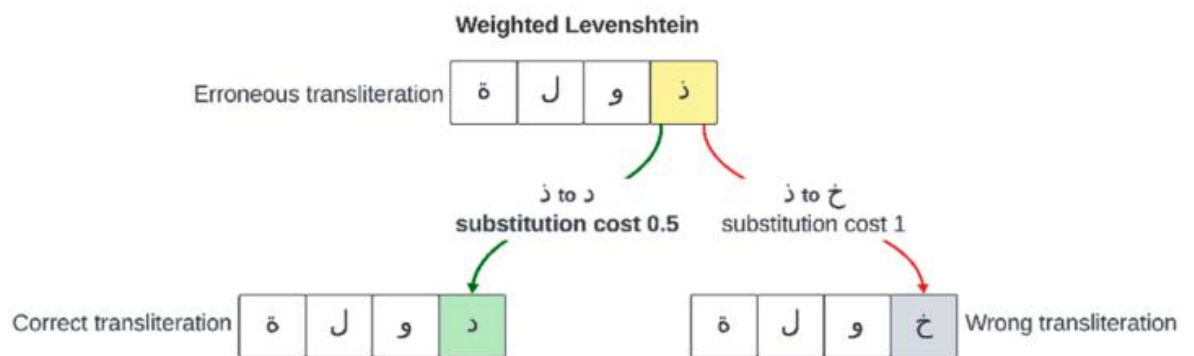


Figure 11 - Weighted Levenshtein method [17]

Dataset

The project intends to create a new fresh dataset, using a mix of data that has been published publicly online, through repositories such as GitHub. The data chosen will be in the category of offensive, non-offensive, hate speech and normal everyday Arabizi text. The data will be cleaned and pre-processed before it is published into the dataset. The data will then be labelled to quickly to identify whether it is deemed offensive or non-offensive. The data will then need to follow the DVLA format of license plates

[19]. That said, multiple variations of each word, will need to be identified to ensure the software cannot be bypassed. This helps reflect the real-world usage of using Arabizi license plates using Latin characters and numbers on vehicles within the DVLA.

Toolkit

This project will investigate multiple toolkits that are open-source and publicly available online. The first toolkit that will be investigated is Hugging Face [20], this is where models such as Arabert [21], DarijaBERT [22] will be explored. Subsequently, Pytorch [23] will look at fine-tuning the dataset on both NLP models. Following this, a public dataset [24] will be chosen to apply pre-processing steps using Python [25] and Pandas [26]. Finally Seaborn [27] will be used to present high level graphs to show the results of the quantifiable metrics of the two NLP models.

Motivation for this study

The motivation for this study is to address a gap where Arabizi can pose a challenge for filtering inappropriate license plates in systems such as DVLA. The current system is focused on English, and this project looks at detecting Arabic, using Arabizi which is written using Latin characters and numbers to fill the gap. A need for the project is that native Arabic speakers, may be offended if license plates are approved.

Methodology

The research methodology will begin with a thorough review of literature to understand the current approach in detecting offensive and non-offensive language in Arabizi,

through natural language processing models. The second stage will look at creating a new dataset, developing the software and evaluating how well the models performed.

This study will follow the research onion approach figure 5. A positivism philosophy is based on empirical evidence and data that can be measurable. The deductive approach will test the hypothesis on whether DarijaBERT-arabizi can outperform Arabert based on inappropriate Arabizi DVLA plates. The strategy will be an experiment to compare each NLP model. A mono method is applied using quantifiable data. The data involves a cross-sectional of collecting the tested data and analysing it within the same period.

The data collection will involve the creation of a new dataset, containing Arabizi formatted license plates that have offensive and non-offensive plates. The NLP model will then be fine-tuned and trained on the dataset. Quantifiable data will be derived from the results to draw conclusions.

The software implementation will use Python [25], Hugging Face transformers [20] and PyTorch [23]. NLP existing models will be imported from Hugging Face website [22] [21]. A new dataset will be created and include Arabizi DVLA license plates for NLP models to train on. The pre-trained NLP models are then fine-tuned and retrained on the same dataset to achieve quantifiable metrics. Every model will then have metrics derived that are testable to draw conclusions and therefore prove/disprove the hypothesis within the dissertation.

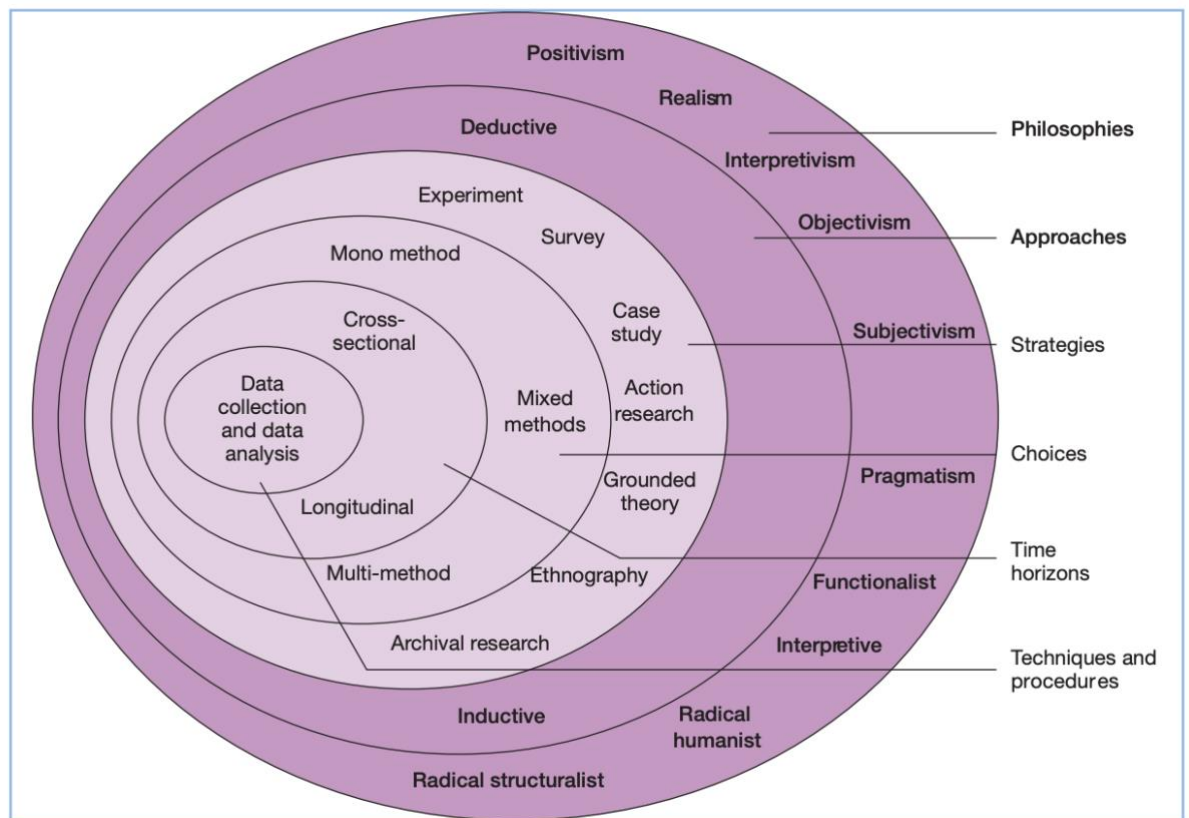


Figure 12 - The research onion [28]

Gantt Chart

The Gantt Chart is shown in figures 6, 7 and 8. This is the current plan the project will be estimated to take, however the timeline for each stage may be subject to change due to unforeseen delays.

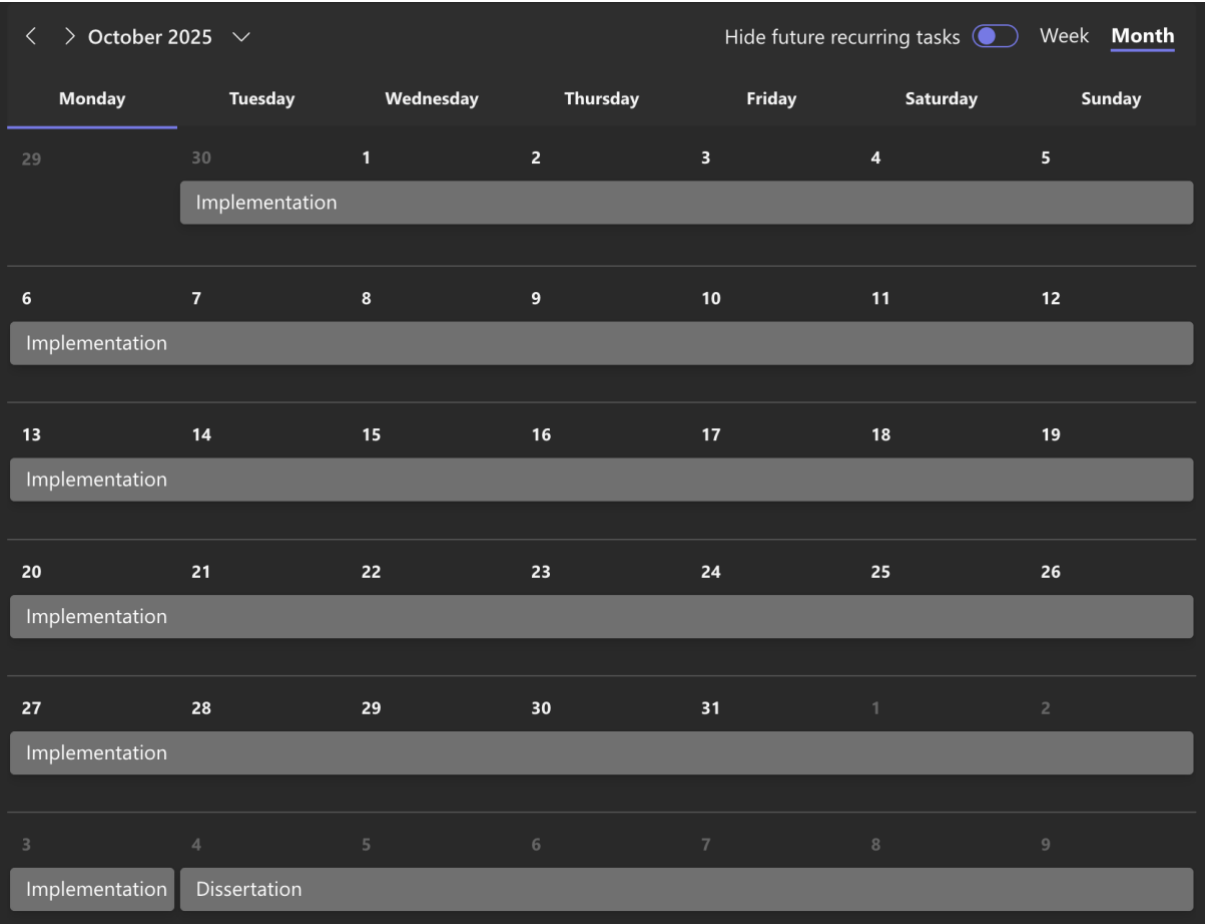


Figure 13 - October Gantt Chart

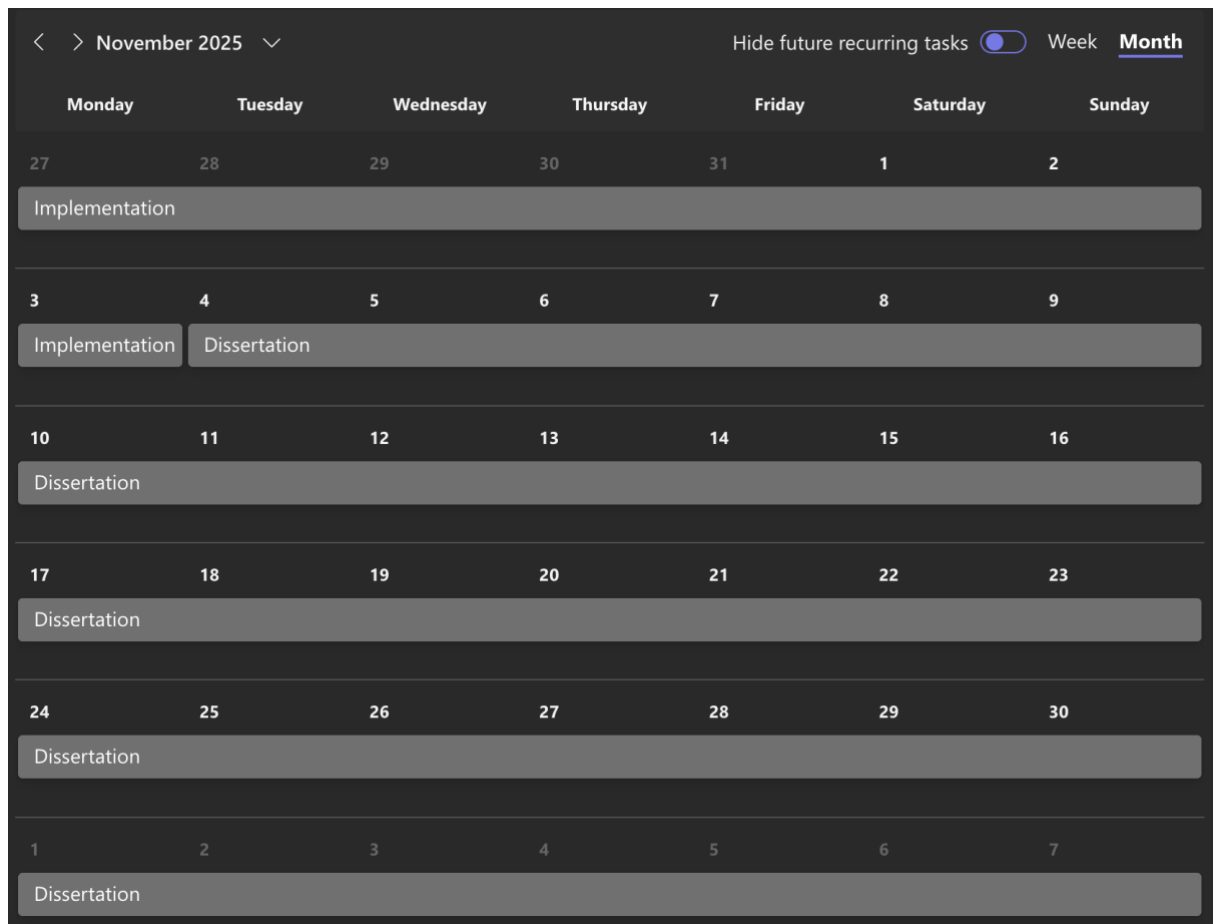


Figure 14 - November Gantt Chart

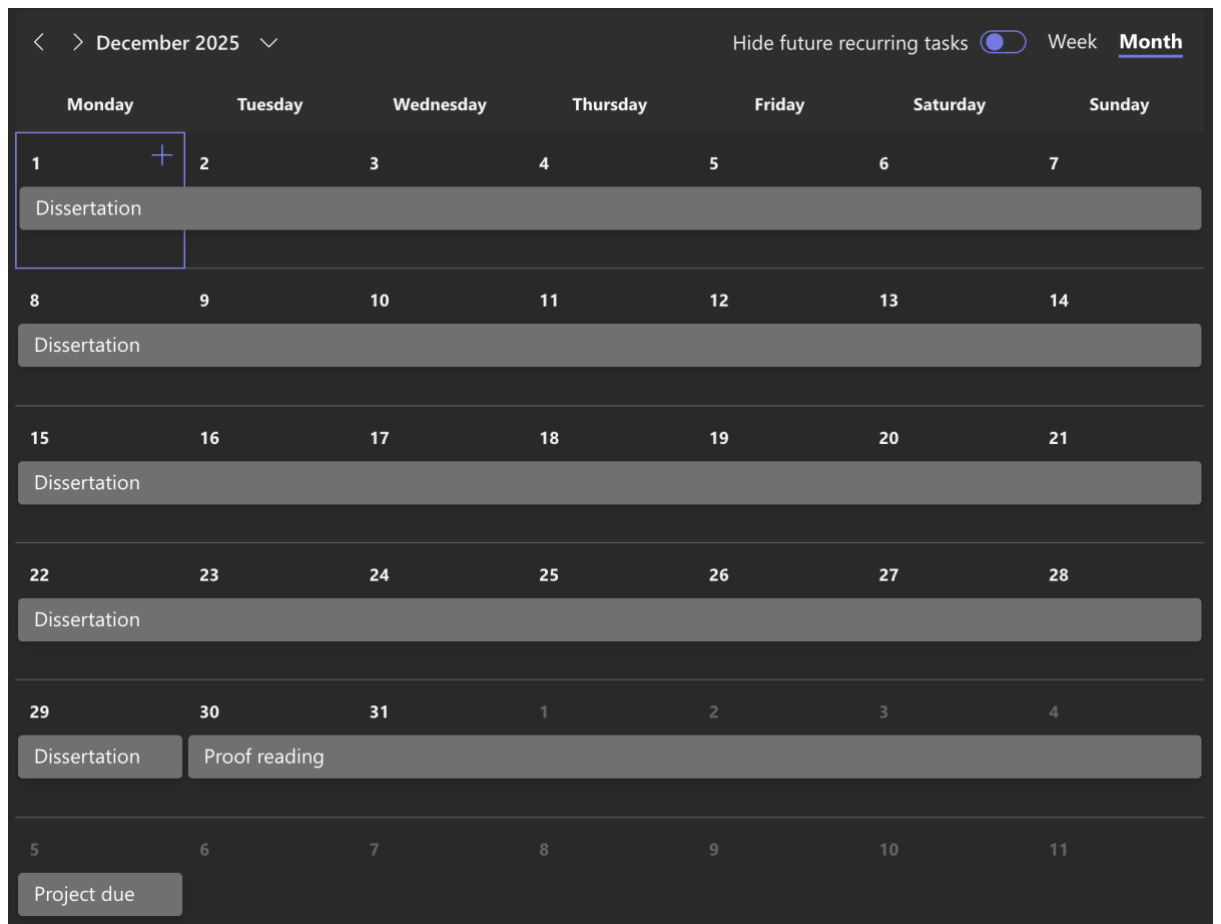


Figure 15 - December Gantt Chart

Project planning

Development methods

The project development method will use the Research Driven Development (RDD) methodology. The software will be developed to support an experiment and analysis based on the hypothesis.

Dissertation

The dissertation is the most important and final written piece of work for the project. The purpose of the dissertation is to provide the analysis and draw conclusions based on the derived results of the testable artefact of the software. The results that will be measured is the accuracy, precision, recall and inference time of the NLP model based on the dataset. An abstract, introduction, thorough review of literature will take place, along with a methodology which will contain the research onion, results and findings, conclusion will be the main parts of the dissertation.

Ethical

Due to GDPR, the project will not use live data and instead will use made up license plates using different variations of license plates using Arabizi to replicate a license plate using the DVLA format.

Expected outcomes / deliverables

The expected outcome is to develop software that can prove the hypothesis by deriving results based on the outcome of the software.

Deliverables

- Project proposal
- Implementation
- A complete dataset with offensive and non-offensive license plates.
- Dataset source code and repository link
- Source code for the software created.
- Infographics for data visualisations
- Drawing quantifiable data from the software
- Video demo of the software and running the experiment.
- Screenshots of data experiment.
- Documentation such as a Readme file
- Completed logbook
- Dissertation

Resources

Academic support

- Weekly meetings with project supervisor
- Learning support 1:1 weekly meeting to provide support and proof-reading.
- Access to UWTSD library portal
- Access to IEEE and ACM portals
- UWTSD Moodle
- Project logbook

Hardware

- MacBook Pro with 32GB Ram and 512GB SSD storage space.

Software

- Git
- GitHub
- Hugging face transformers
- Pytorch
- Python
- Pandas
- Seaborn
- OBS (video recording)
- Microsoft Word
- Microsoft planner
- Zotero (reference manager)

Conclusion

To conclude this project proposal, the project aims to provide the reader with the necessary information to understand the purpose of the project. This submission acts as a proof of concept in implementing a solution within license plates to detect inappropriate number plates using NLP. The findings of the project will provide license plate moderation for future development and contribute to strengthening the field of NLP. Future work may contribute to creating the software to support further languages that support the use of Latin characters and numbers as well as expanding the dataset to contain more variations of offensive and non-offensive license plates to help the NLP model be trained on.

Bibliography

- [1] A. B. Sadallah, O. Ahmed, S. Mohamed, O. Hatem, D. Hesham, and A. H. Yousef, ‘ANER: Arabic and Arabizi named entity recognition using transformer-based approach’, in *2023 Intelligent Methods, Systems, and Applications (IMSA)*, IEEE, 2023, pp. 263–268.
- [2] O. Amezian, S. Hajbi, and Y. Chihab, ‘Towards a large Biscrit Moroccan Lexicon’, in *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, IEEE, 2023, pp. 1–6.
- [3] O. Amezian, S. Hajbi, Z. S. Houssaini, and Y. Chihab, ‘Training an LSTM-based Seq2Seq model on a Moroccan biscrit lexicon’, in *2023 9th International Conference on Optimization and Applications (ICOA)*, IEEE, 2023, pp. 1–6.
- [4] I. A. Alsharif, ‘Analysis of Sentiment and Sarcasm in Arabic YouTube Comments Utilizing AraBERT’, in *2025 2nd International Conference on Advanced Innovations in Smart Cities (ICAISC)*, Jeddah, Saudi Arabia: IEEE, Feb. 2025, pp. 1–5. doi: 10.1109/ICAISC64594.2025.10959435.
- [5] S. Das and D. Das, ‘Natural Language Processing (NLP) Techniques: Usability in Human-Computer Interactions’, in *2024 6th International Conference on Natural Language Processing (ICNLP)*, 2024, pp. 783–787. doi: 10.1109/ICNLP60986.2024.10692776.
- [6] J. Francis and M. Subha, ‘An Overview of Natural Language Processing (NLP) in Healthcare: Implications for English Language Teaching’, in *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Kirtipur, Nepal: IEEE, Oct. 2024, pp. 824–827. doi: 10.1109/I-SMAC61858.2024.10714890.
- [7] Amazon, ‘Alexa’, Alexa. [Online]. Available: <https://alexa.amazon.co.uk/>

- [8] Apple, ‘Siri’, Siri. [Online]. Available: <https://www.apple.com/siri/>
- [9] Google, ‘Google Assistant’, Google Assistant. [Online]. Available: <https://assistant.google.com/>
- [10] S. Ravichandiran, *Getting started with Google BERT: build and train state-of-the-art natural language processing models using BERT*. Birmingham Mumbai: Packt, 2021.
- [11] N. Mossad, Y. Mohamed, A. Fares, and A. B. Zaky, ‘Arabic text sentiment analysis and emotion classification using transformers’, in *2023 11th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC)*, Alexandria, Egypt: IEEE, Dec. 2023, pp. 131–137. doi: 10.1109/JAC-ECC61002.2023.10479609.
- [12] Y. Liu, H. Huang, J. Gao, and S. Gai, ‘A study of Chinese Text Classification based on a new type of BERT pre-training’, in *2023 5th International Conference on Natural Language Processing (ICNLP)*, Guangzhou, China: IEEE, Mar. 2023, pp. 303–307. doi: 10.1109/ICNLP58431.2023.00062.
- [13] A. Kerkri, M. A. Madani, H. Ghouch, and K. E. Majdoub, ‘Arabic Web Page Classification with AraBert and Mixture vectorization Methods: Analysis Using K-NN, LSTM and Naive Bayes’, in *2023 Fourth International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, Kuwait, Kuwait: IEEE, Oct. 2023, pp. 43–47. doi: 10.1109/IDSTA58916.2023.10317860.
- [14] Z. Boulouard, M. Ouaisa, M. Ouaisa, M. Krichen, M. Almutiq, and K. Gasmi, ‘Detecting Hateful and Offensive Speech in Arabic Social Media Using Transfer Learning’, *Appl. Sci.*, vol. 12, no. 24, p. 12823, Dec. 2022, doi: 10.3390/app122412823.
- [15] M. El-Ramly *et al.*, ‘CairoDep: Detecting Depression in Arabic Posts Using BERT Transformers’, in *2021 Tenth International Conference on Intelligent*

- Computing and Information Systems (ICICIS)*, Cairo, Egypt: IEEE, Dec. 2021, pp. 207–212. doi: 10.1109/ICICIS52592.2021.9694178.
- [16] E. S. Alshahrani and M. S. Aksoy, ‘Adversarially Robust Multitask Learning for Offensive and Hate Speech Detection in Arabic Text Using Transformer-Based Models and RNN Architectures’, *Appl. Sci.*, vol. 15, no. 17, p. 9602, Aug. 2025, doi: 10.3390/app15179602.
- [17] S. Hajbi, O. Amezian, N. E. Moukhi, R. Korchiyne, and Y. Chihab, ‘Moroccan Arabizi-to-Arabic conversion using rule-based transliteration and weighted Levenshtein algorithm’, *Sci. Afr.*, vol. 23, p. e02073, Mar. 2024, doi: 10.1016/j.sciaf.2024.e02073.
- [18] A. Masmoudi, M. E. Khmekhem, M. Khrouf, and L. H. Belguith, ‘Transliteration of Arabizi into Arabic Script for Tunisian Dialect’, *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 19, no. 2, pp. 1–21, Mar. 2020, doi: 10.1145/3364319.
- [19] Department for Transport, ‘Displaying number plates’, GOV.UK. [Online]. Available: <https://www.gov.uk/displaying-number-plates>
- [20] Hugging Face, ‘The AI community building the future.’, Hugging Face. [Online]. Available: <https://huggingface.co/>
- [21] W. Antoun, F. Baly, and H. Hajj, ‘AraBERT NLP model’, AraBERT v1 & v2 : Pre-training BERT for Arabic Language Understanding. [Online]. Available: <https://huggingface.co/aubmindlab/bert-base-arabertv02>
- [22] K. Gaanoun *et al.*, ‘SI2M-Lab/DarijaBERT-arabizi’, Hugging Face. [Online]. Available: <https://huggingface.co/SI2M-Lab/DarijaBERT-arabizi>
- [23] ‘PyTorch Getting Started’, Get Started. [Online]. Available: <https://pytorch.org/>

- [24] I. Bensalem, M. Ait, Meryem, and P. Rosso, 'Arabizi offffensive langauge'.
GitHub, Online. [Online]. Available: [https://github.com/Imene1/Arabizi-offensive-
language/tree/main](https://github.com/Imene1/Arabizi-offensive-language/tree/main)
- [25] Python Software Foundation, 'Python', Python. [Online]. Available:
<https://www.python.org/>
- [26] Pandas, 'Pandas'. [Online]. Available: <https://pandas.pydata.org/>
- [27] Michael Waskom, 'Seaborn', seaborn: statistical data visualization. [Online].
Available: <https://seaborn.pydata.org/index.html>
- [28] M. Saunders, P. Lewis, and A. Thornhill, *Research methods for business students*, 4. ed. Harlow: Financial Times Prentice Hall, 2007.