



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	3
Hypothesis.....	3
Project Aim	3
Objectives:	4
Literature review.....	6
Arabizi	6
Natural Language Processing (NLP)	7
BERT	7
ARABERT	8
Offensive language in Arabic	8
Preprocessing and normalisation techniques	11
Dataset	14
Toolkit.....	15
Motivation for this study	15
Methodology.....	15
Gantt Chart.....	18
Project planning	21
Development methods	21
Dissertation	21
Ethical	21
Expected outcomes / deliverables	22
Deliverables	22
Resources	23
Conclusion.....	24
Bibliography	25

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 1- 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 2 - 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 3 - 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 4 - 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 Levenstein 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].

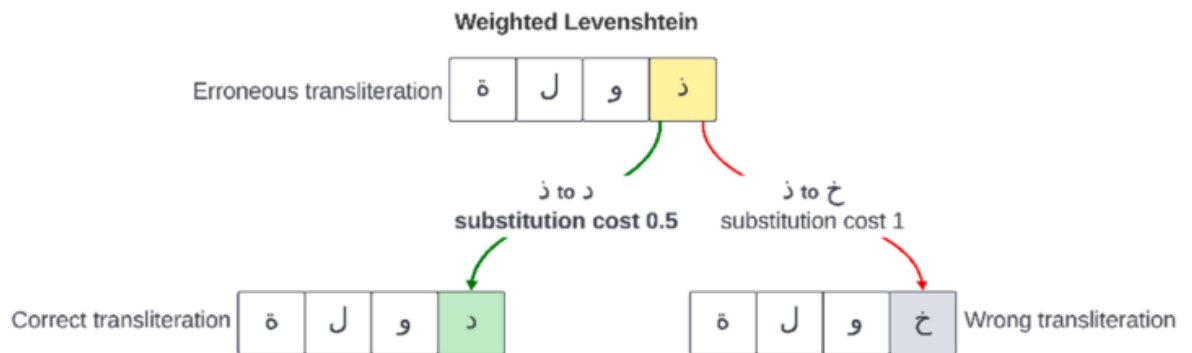


Figure 1 - Weighted Levenshtein method [17]

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.

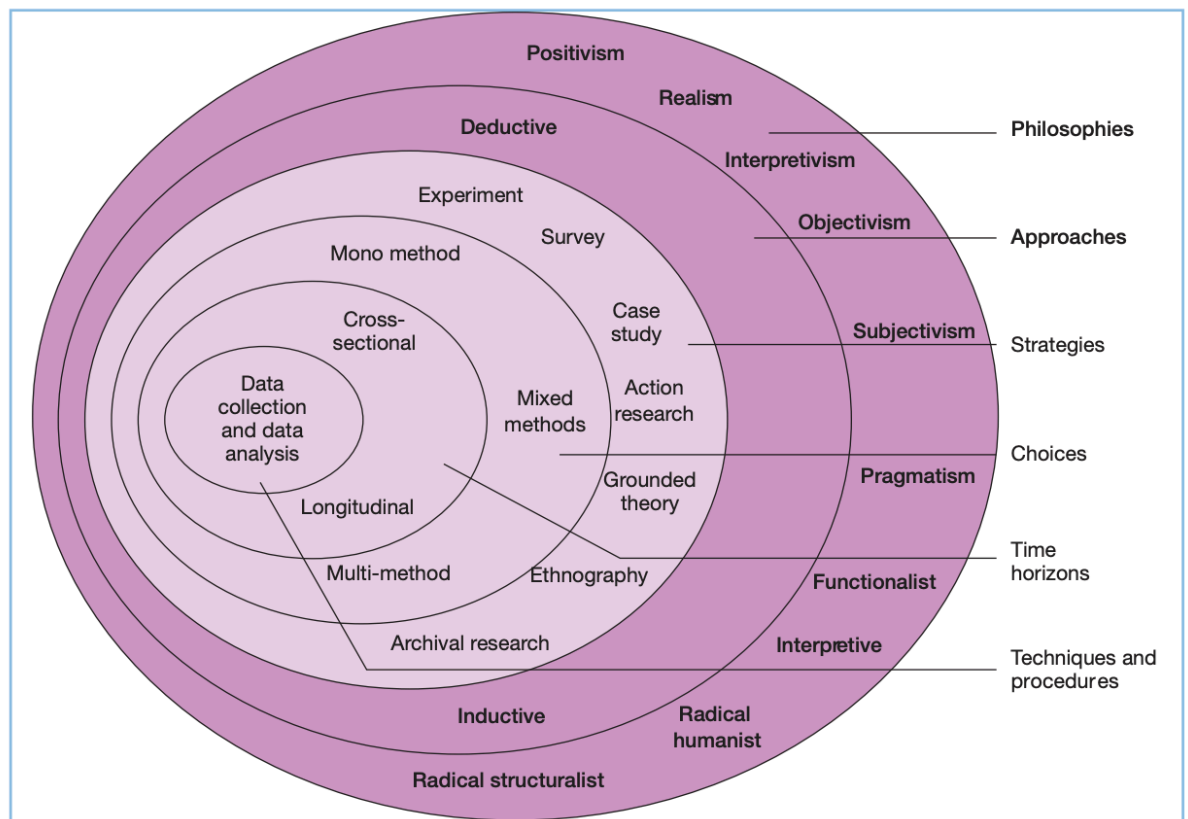


Figure 2 - The research onion [28]

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.

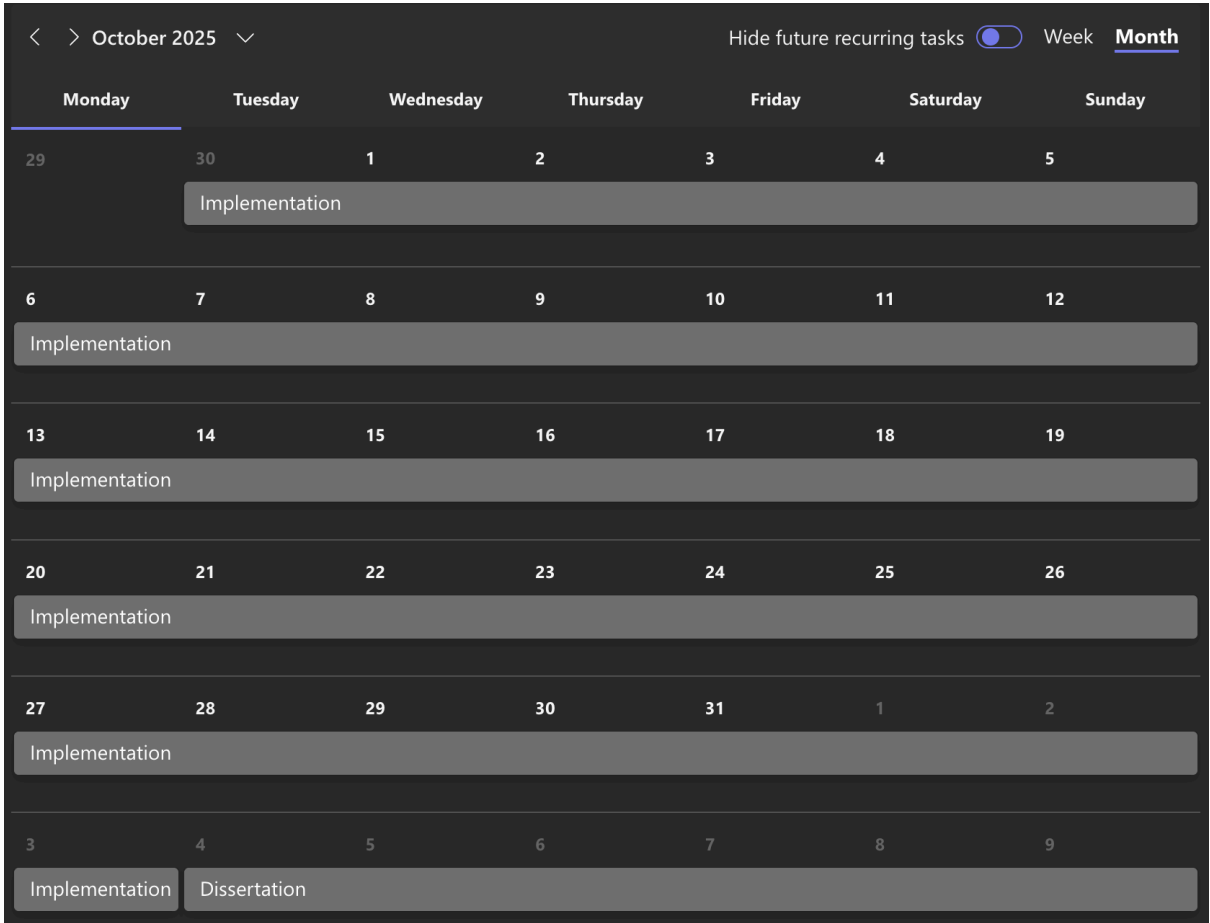


Figure 3 - October Gantt Chart

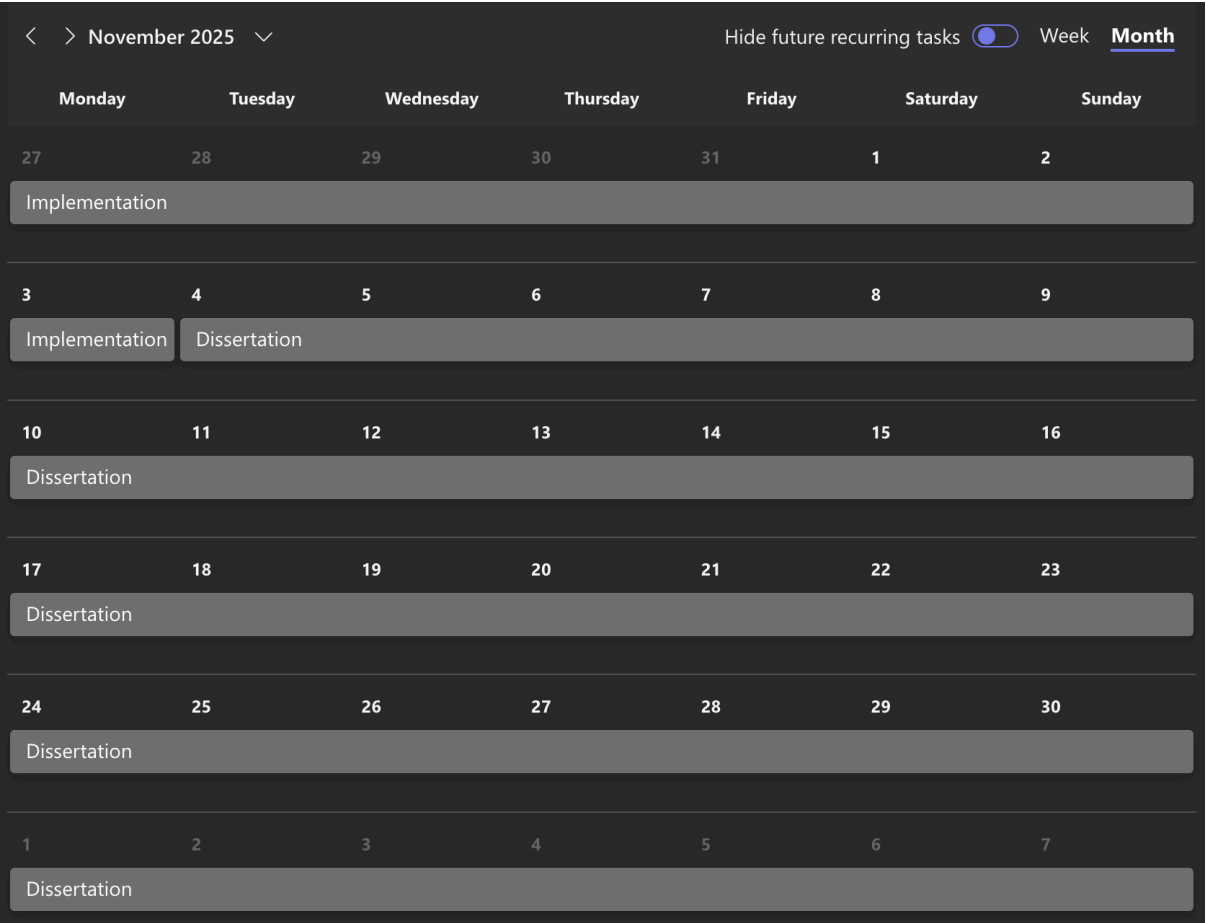


Figure 4 - November Gantt Chart

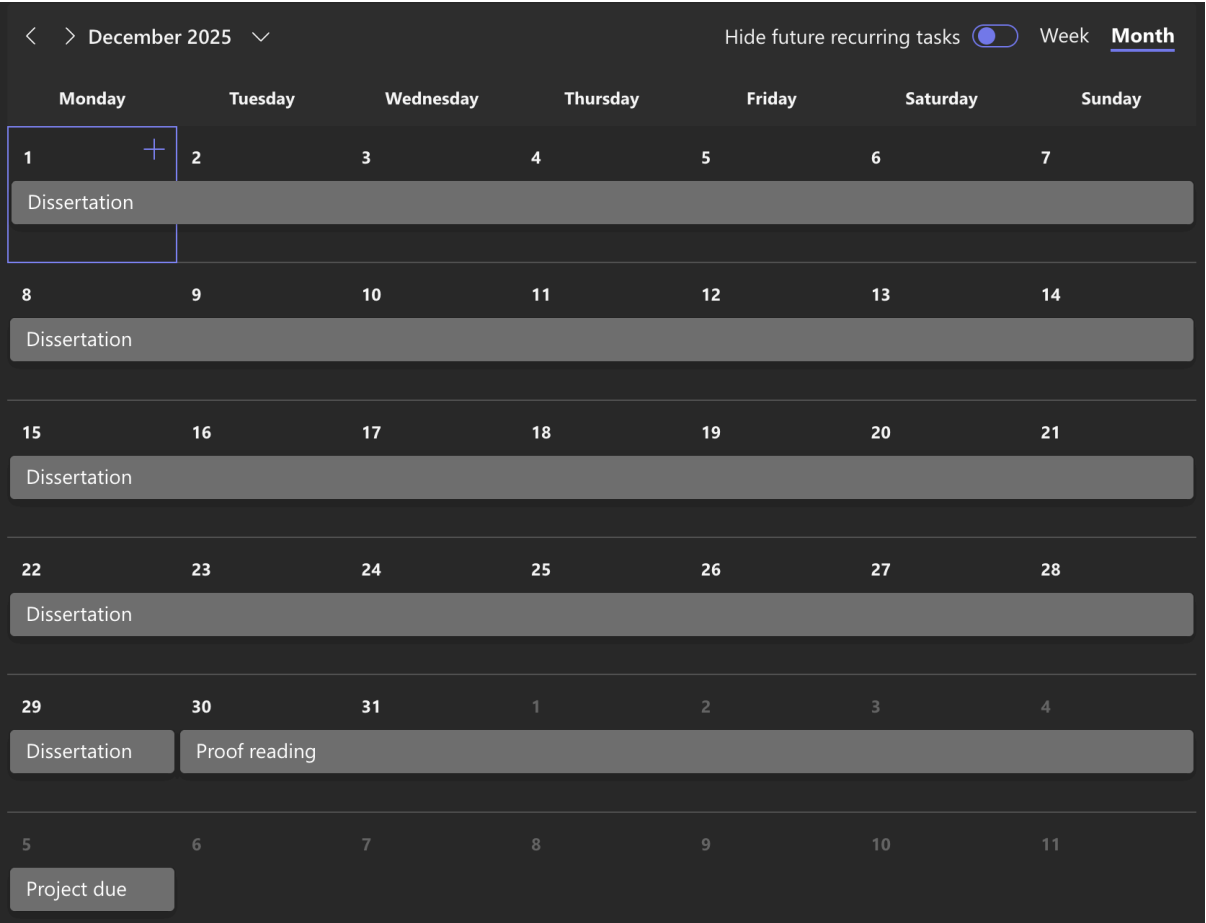


Figure 5 - December Gantt Chart

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 biscript 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 offensive language’. GitHub, Online. [Online]. Available: <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.