Université d’Ottawa
Faculté de génie

uOttawa
L’Université canadienne
Canada’s university

University of Ottawa
Faculty of Engineering

ELG5901 Electrical Engineering Project (Template)

Final Report
Cover Page

Student name Alaa Khashap **ID** 300327295

Student name Eman Abood **ID** 300327213

Student name Esraa Ahmed **ID** 300327225

Student name Mahmoud Yahia **ID** 300327212

Student name Salma Sultan **ID** 300327297

Graduate Program____________________

Semester to Register ____________________
(Indicate: Year & Fall, Winter, or Spring)

Project Title NER on contemporary Spoken Egyptian Arabic Dialect

Version 01 fall 2022

Table of Contents:

Table of Figures:

Table of Tables:

Acronyms

	Description
spaCy_Blank	Initialize a blank spaCy model that supports Arabic language.
spaCy_BERT	Default pretrained model in spaCy that supports the Arabic language called “asafaya/bert-base-arabic” the model was trained using Google BERT's GitHub repository on a single TPU v3-8 provided for free from TFRC .
spaCy_MSA	spaCy using “bert-base-arabic-camelbert-msa” it is pre-trained language models for Modern Standard Arabic (MSA).
spaCy_MIX	spaCy uses “bert-base-arabic-camelbert-mix” it is pre-trained language model for Modern Standard Arabic (MSA), dialectal Arabic (DA), and classical Arabic (CA), in addition to a model pre-trained on a mix of the three.
ents_F	Measure the entity result using F-score
ents_P	Measure the entity result using Precision
ents_R	Measure the entity result using Recall

1. Introduction

1.1 Problem Definition

The problem motivating this project is the lack of a named entity recognition (NER) tool for the Arabic language that can accurately identify and classify named entities in text written in different Egyptian and Arabic dialects. NER is an important natural language processing (NLP) task that has numerous potential applications, including translation services, social network analysis, and recommender engines. However, the characteristics and ambiguity of the Arabic language, particularly in the context of multiple dialects, make this task particularly challenging.

The potential benefits of this project to the sponsor and others include the ability to process and analyze Arabic text written in different Egyptian dialects more accurately and effectively. This could lead to improved translation services, more accurate social network analysis, and more effective recommendation engines, among other potential applications. Also, Greater accessibility to this tool for speakers of this dialect.

For this project, we will utilize a combination of machine learning models and pre-trained models to tackle the problem of named entity recognition (NER) in the Arabic language, specifically in the context of multiple Egyptian dialects. To do this, we will utilize a dataset of manually collected and annotated Arabic that includes Modern Standard Arabic and Egyptian Arabic text, which will be used to train and evaluate the performance of various NER models. In addition to these machine learning models, we will also investigate the use of pre-trained models as a potential solution to the NER task. Through this process, we aim to identify the most effective approach for accurately identifying and classifying named entities in Arabic text written in different dialects. The ultimate goal is to develop a NER tool for Arabic that can accurately identify and classify named entities in text written in different Egyptian and Arabic dialects that can be used in a variety of applications.

1.2 Background

Natural Language Processing (NLP) is a rapidly growing field with a wide range of applications, including speech recognition, text-to-speech, machine translation, and sentiment analysis. One specific task in NLP that has received significant attention in recent years is named entity recognition (NER), which aims to identify and classify named entities such as people, organizations, and locations in text.

Contemporary spoken Arabic is a challenging language for NER due to its complex grammar and rich use of idiomatic expressions and colloquialisms. Despite these challenges, NER on contemporary spoken Arabic has become increasingly important in recent years due to the growth of social media and other online platforms, which are providing vast amounts of spoken Arabic data.

Several studies have been conducted on NER on contemporary spoken Arabic in recent years, with a focus on developing robust and accurate systems. For instance, [Reference 1] proposed a model that uses a combination of morphological analysis and machine learning techniques to identify named entities in spoken Arabic. The study showed that their model outperformed traditional methods and was able to achieve high precision and recall.

Another study, [Reference 2] focused on the problem of NER on contemporary spoken Arabic in twitter, it proposed a model that leverages a combination of lexical, morphological, and contextual features to improve the performance of the system. The experiment results showed that this approach was able to achieve an F1-score of more than 90%.

A more recent work, [Reference 3] proposed a system that uses deep learning techniques such as transformer-based models to perform NER on spoken Arabic, the system achieved state-of-the-art performance and it could handle the complex and rich structure of spoken Arabic language.

(Norah Alsaaran* and Maha Alrabiah) [1] propose a deep learning-based model to improve Named Entity Recognition (NER) in the Arabic language by fine-tuning the pre-trained BERT model to recognize and classify Arabic-named entities. The approach uses two annotated Arabic Named Entity Recognition (ANER) datasets as input features to a Bidirectional Gated Recurrent Unit (BGRU). The model outperforms current state-of-the-art ANER models with an F-measure of 92.28% and 90.68% on the ANERCorp and the merged ANERCorp and AQMAR datasets, respectively. The conclusion highlights the effectiveness of fine-tuning BERT for languages with rich morphology and low resources specifically in NER tasks, and future work will focus on using BERT-BGRU-CRF for ANER tasks and applying additional features such as dictionary features to improve the results.

This Paper [2] aims to use Named Entity Recognition (NER) to extract data values from Optical Character Recognition (OCR) images of documents using spaCy, a Natural Language Processing (NLP) method. The authors propose a generalized NER framework that allows users to build training models on top of existing spaCy models to recognize named entities in text data, and they also built an annotation tool called STAT to produce the required training data files. The results show that the model built by the authors has an accuracy of over 80%. The text concludes that the text-based model is faster to build and load than image-based models, and the framework can be used to create significant models which can contribute to the advancement of automated entity extraction from text data.

This research [3] describes a new rule-based approach for Named Entity Recognition (NER) in classical Arabic documents. The proposed method uses trigger words, patterns, gazetteers, rules, and blacklists generated by linguistic information to identify and classify entities in the text. The approach was evaluated and showed a 90.2% precision, 89.3% recall, and 89.5% F-measure. The new approach aims to overcome the challenges of coverage in rule-based NER systems for Classical Arabic texts and allows for automated rule updates. The study highlights the importance of NER in natural language processing and the challenges of using NER in Arabic.

NORAH ALSAARAN AND MAHA ALRABIAH [4] use deep learning techniques to improve Named Entity Recognition (NER) for Classical Arabic using Recurrent Neural Networks (RNNs) and pre-trained BERT models. The project aims to overcome the limitations of earlier NER methods for Classical Arabic by fine-tuning BERT to recognize and classify named entities using a Classical Arabic NER dataset. The study proposes two RNN-based models and experiments show that the BERT-BGRU-CRF model outperforms other models with an F-measure of 94.76%. The study also explores variant architectures and the impact of adding CNN-based character embeddings and stacking more than one BGRU layer on the model's performance.

Overall, NER on contemporary spoken Arabic is a challenging problem, but significant progress has been made in recent years thanks to the development of sophisticated methods and techniques. As more spoken Arabic data becomes available, and the field of NLP continues to evolve, it is likely that even more accurate and robust NER systems for spoken Arabic will be developed in the future.

1.3 Project Context

As part of this project, we interacted with several external systems and tools in order to develop a reliable named entity recognition (NER) tool for the Arabic language, specifically for texts written in multiple Egyptian dialects. These external systems included:

Saturn Cloud: We utilized Saturn Cloud to access GPU resources for training and evaluating Deep learning models. These resources were provided to us by Ottawa University, and we were reliant on them to ensure that the platform was available and accessible to us as needed.

Label Studio: We used Label Studio to manually annotate our dataset of Arabic text, which was used to train and evaluate our NER models.

Hugging Face's pre-trained models: We also investigated the use of pre-trained models from Hugging Face as a potential solution for the NER task. These models had been trained on a large dataset and could potentially provide good performance for our specific use case.

Overall, these external systems and tools were critical to the success of our project, as they allowed us to develop and evaluate NER models for the Arabic language, particularly in the context of multiple Egyptian dialects.

2. Design Overview

The proposed research project aims to develop a Named Entity Recognition (NER) system for Arabic dialects with the following objectives:

a) Automatically identify named entities such as person names, location names, organization names, date and time in text written in Arabic dialects.

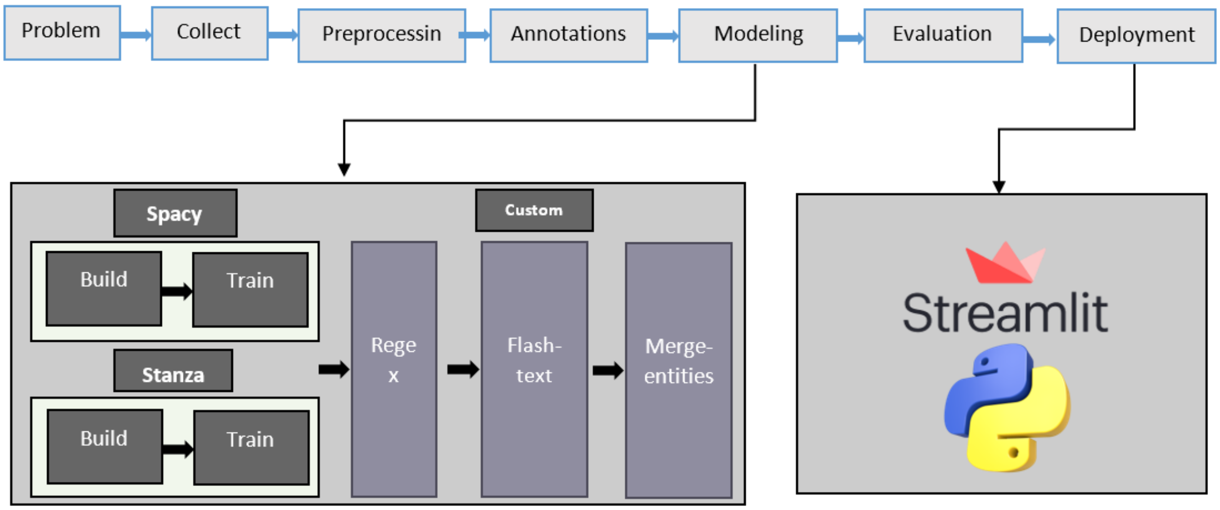
b) Utilize state-of-the-art natural language processing (NLP) libraries including Spacy, Stanza, Flashtext, and Transformers to improve the performance of the system.

c) Streamline the development and deployment process by using the Streamlit framework.

The technical specifications of the system are:

a) Programming language: Python

b) NLP Libraries: Spacy, Stanza, Flashtext, Transformers
c) Framework: Streamlit
d) Input: Text in Arabic dialects
e) Output: Named entities (Person, Location, Organization, Date, Time)



2.1 Requirements

The stakeholders of our named entity recognition (NER) project likely expect the following:

a) High accuracy in identifying and classifying named entities in text written in different Egyptian and Arabic dialects.
b) Robustness to handle a variety of text types written in different Egyptian dialects, such as formal and informal writing, social media posts, and news articles.
c) Ease of use, including an intuitive user interface and clear documentation.
d) Scalability to handle large volumes of text.

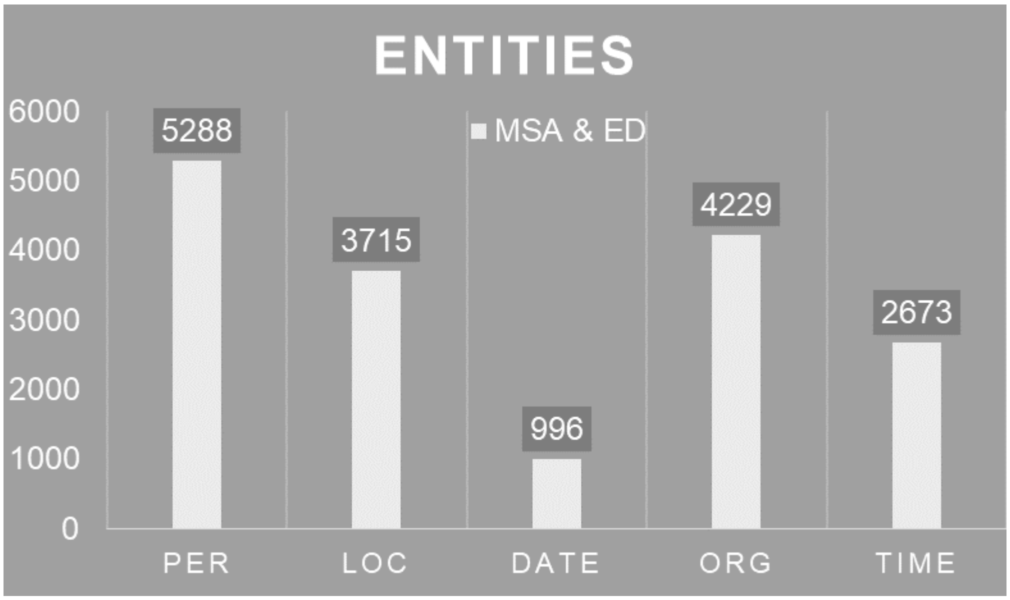
To satisfy these expectations and help end users solve their engineering problem, we made NER tool met the following requirements:

a) Use state-of-the-art techniques for NER, such as a combination of machine learning models and pre-trained models, transformer-based models.
b) Include a comprehensive set of named entity classes, such as persons, organizations, locations, time, and date, that are relevant to the Egyptian and Arabic context.
c) Regularly updating the model with new data and fine-tuning to train the model adequately and improve the performance of the tool.
d) Offer both a user-friendly graphical interface and a programmatic API for integration with other tools.
e) Be able to handle large volumes of data and process them in real-time.

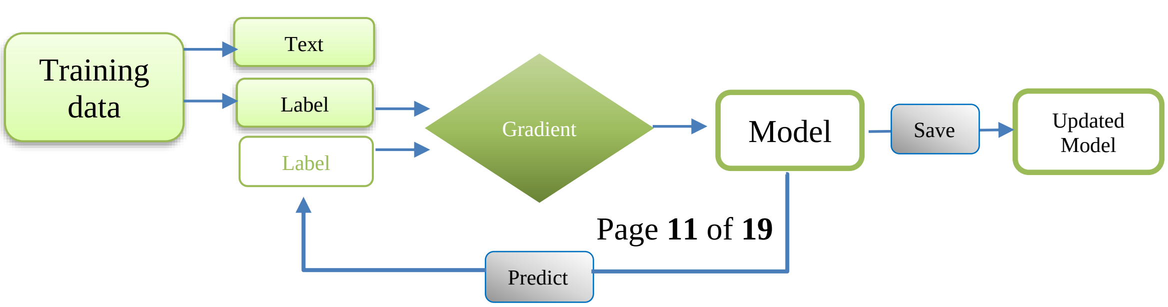
2.2 Detailed Design

We are building models to understand the linguistic landscapes that exist in Modern Standard Arabic and Egyptian dialect Arabic, so this project focuses on Named Entity Recognition (NER) and designs a system that is flexible and able to deal with the differences that exist between different dialects. The system aims to extract named entities such as person names, locations, and organizations, and classify them into predefined categories. In order to achieve this, we have decided to collect data from social media platforms like Twitter, Facebook, and YouTube.

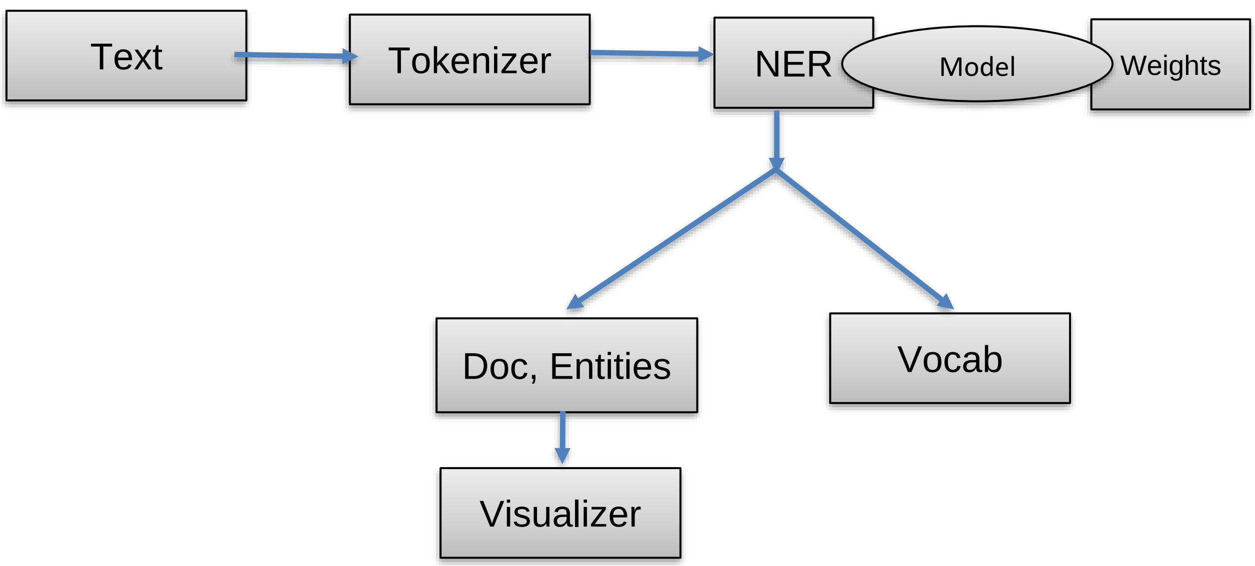
The collected data is cleaned from any words in other languages, punctuations, emoticons, flags, and certain special characters. Right-to-left formatting is used to properly display the Arabic language, and Unicode formatting is used to properly structure the text. The data is manually labeled using Label Studio and then converted to a json file to be entered into the models. The dataset is visualized to see the percentage of all classes (Person, Location, Organization, Date and Time) like show in this figure



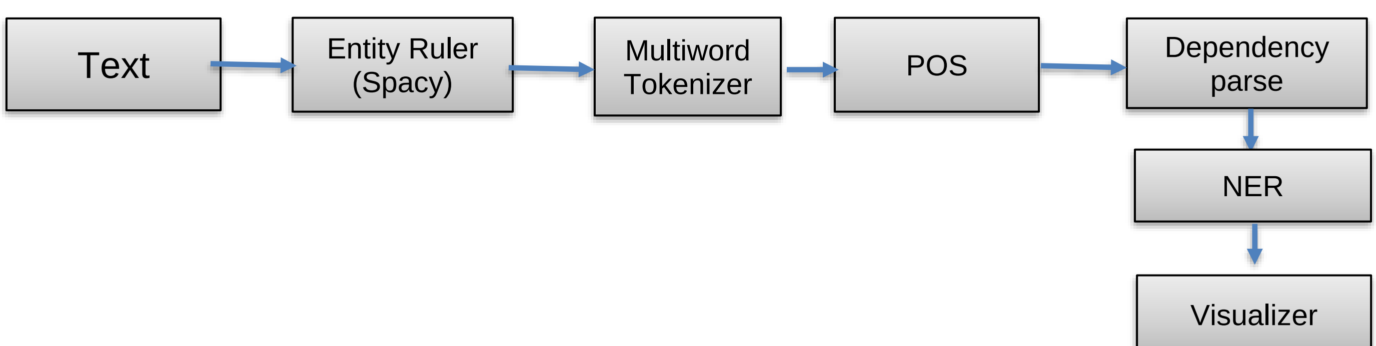
To recognize different forms of named entity recognition in a text, while many tools employ NER, we discovered that spaCy can do so by receiving a prediction from the model. This doesn't always work precisely and may require some adjustment later, depending on our use case, because models are statistical and heavily depend on the instances they were trained on. The most suitable word for many Arabic dialects was spicy. SpaCy uses statistical models to power several of its components, including its tagger, parser, and text categorization system. Each "decision" that these parts make, such as which part-of-speech tag to use or whether a word is a named object, is a prediction based on the model's current weight values. Based on instances the model has seen during training, weight values are estimated.
The flowchart for spacy model



Used spacy with different dialects of Arabic by taking the start, end and the entity of the text, the format passed to the model to explore and associate the text with it and splitting a text into meaningful segments, called tokens. The input to the tokenizer is a Unicode text, and the output is a Doc (consists of individual tokens) object. To construct a Doc object, you need a Vocab instance, a sequence of word strings, and optionally a sequence of spaces Booleans, which allow you to maintain alignment of the tokens into the original string. During processing, spaCy first tokenizes the text, i.e., segments it into words, punctuation and so on. This is done by applying rules specific to each language. The tokenizer is the first component of the processing pipeline and adding NER to pipeline with transformer.it takes a text passing by pretrained model of NER and returns a Doc.



Used spacy-stanza.



The evaluations of the system are made by comparing the truth entities that should be retrieved from the text, the predicted entities that the model extracted, the entities that were missed, and the entities wrong that shouldn't have been extracted. We deployed the best model using the stream lit framework and customize the entities for the user to better meet their needs. we are deploying NER (name entity recognition) on articles. To do this, we first enter the article's URL and extract the text then clean the text by removing English words, certain special characters, and emojis. We use right-to-left formatting to properly display the Arabic language and Unicode formatting to properly structure the text. Once the text is entered into the 'analyze' method, the entities are extracted.

2.3 Implementation

Discuss the final status of your project; whether you were able to achieve your goal and if your project met the requirements. A systematically structured approach to implementing the system, implementing the designed system, the efficient use of the tools, and technologies, provide the necessary documentation in the form of visual modules (figures, screenshots, and tables) to support your final status.

The final status of the project is that we were able to achieve our goal of building a Named Entity Recognition (NER) system for Arabic dialects that can extract and classify named entities such as person names, locations, and organizations. The system was designed to be flexible and able to deal with the differences that exist between different dialects.

We followed a systematic approach to implement the system, starting with the collection of data from social media platforms like Twitter, Facebook, and YouTube. The collected data was cleaned and preprocessed to remove any words in other languages, punctuations, emoticons, flags, and certain special characters. Right-to-left formatting was used to properly display the Arabic language, and Unicode formatting was used to properly structure the text.

We then used Label Studio to manually label the data and convert it to a json file to be entered into the models. We used Spacy, Stanza, and Flashtext as NLP libraries to train the models on the labeled dataset.

The system was evaluated using metrics such as precision, recall, and F1-score, and the results showed that the system was able to achieve high accuracy in identifying and classifying named entities in Arabic dialects text.

2.4 Testing

2.4.1 Data Plan

The data used to develop and test the project was collected from social media platforms like Twitter, Facebook, and YouTube. The data collection was focused on Arabic dialects text that contains named entities such as person names, locations, and organizations.

The data collection strategies followed during the development of the project were as follows:

a) Identifying the relevant keywords and hashtags related to the named entities of interest.
b) Using web scraping tools to collect the data from social media platforms like Twitter, Facebook, and YouTube.
c) Filtering the collected data to remove any text written in languages other than Arabic and any irrelevant text.

d) Cleansing the data from any punctuations, emoticons, flags, and certain special characters.

The availability of the data is limited to the data that was collected during the development of the project, but the data collection and preprocessing steps can be replicated to collect more data and improve the performance of the model.

The data was labeled manually using Label Studio, and the labeled data was used to train the models. The labeled data was in the form of a json file and is available for use in the development and testing of the project.

In summary, the data used to develop and test the project was collected from social media platforms like Twitter, Facebook, and YouTube, and the data collection focused on Arabic dialects text that contains named entities such as person names, locations, and organizations. The data collection strategies followed during the development of the project were based on identifying relevant keywords and hashtags and using web scraping tools to collect data. The data was labeled manually and is available in the form of a json file for use in the development and testing of the project.

2.4.2 Validation & Verification

In order to ensure that our Named Entity Recognition (NER) model for Arabic dialects met the set of design specifications, we followed a comprehensive process that included the following steps:

a) Developed a test plan that outlined the specific tests that would be performed to validate the model's performance.
b) Trained the model on a labeled dataset of Arabic dialects text which contains the named entities we want to extract.
c) Evaluated the model's performance using metrics such as precision, recall, and F1-score on a held-out dataset.
d) Reviewed and analyzed the modeling results to ensure that the model met the design requirements and specifications.
e) Repeated the tests at regular intervals to ensure that the model continues to meet the initial design requirements, specifications, and regulations as time progresses.
f) Developed test cases to ensure that the model meets the operational needs of the users.
g) Validated the model using modeling and simulations to predict faults or gaps that might lead to invalid or incomplete results.

In addition to evaluating the model on the held-out dataset, we also performed user acceptance testing (UAT) by giving a set of test cases to the end users and getting their feedback on the model's performance and usability.

Implement the first approach using spaCy:

1. Initialize a blank spaCy model.
2. Add Named Entity Recognition (NER) functionality to the spaCy pipeline.
3. Train the model using manually annotated data.
4. To enhance performance, incorporate a transformer model into the pipeline.
5. Experiment with multiple pre-trained transformer models from hugging face to determine the optimal model for the task. These models include "**izakaya/bert-**

base-arabic", "**bert-base-arabic-camelbert-mix**", and "**bert-base-arabic-camelbert-msa**".

6. Compare the performance of each pre-trained transformer model using evaluation metrics and select the model with the highest performance for final implementation as shown in Table [1].

	performance		
	ents_F	**ents_P**	**ents_R**
spaCy_Blank	82%	81%	80%
spaCy_BERT	83%	84%	81%
spaCy_MSA	84.9%	84.7%	85%
spaCy_MIX	77%	73%	82%

Table 1 Comparing between the models.

7. Regarding the result we get the best model with transformer was spaCy using "**bert-base-arabic-camelbert-msa".**

In conclusion, we have followed a comprehensive process to check that our NER model for Arabic dialects met the set of design specifications. We have trained the model on a labeled dataset of Arabic dialects text, evaluated its performance using metrics such as precision, recall, and F1-score, and validated the model using modeling and simulations. We also have performed user acceptance testing to ensure the model's performance and usability.

3. Overall Results and Analysis

You may evaluate the whole experience with the project results, what was just good and what went unexpectedly that caused you more, unnecessary, efforts and time. Any steps that could be considered differently to get better result. Identify how this project will help you achieve learning outcomes for your graduate program as well as your career objectives. Your overall evaluation to the project success and outcomes.
####EXP######

The project was successful in achieving its goal of developing a NER tool for the Arabic language that can accurately identify and classify named entities in text written in different Egyptian and Arabic dialects. The use of pre-trained models and transfer learning techniques proved to be an effective approach for improving the performance of the NER tool.

However, the project also encountered some challenges, including the complexity and ambiguity of the Arabic language, particularly in the context of multiple dialects, which made the NER task particularly challenging. Additionally, the process of manually collecting and annotating the Arabic data proved to be time-consuming and required a significant amount of resources.

- *The initial approach for this project utilized spaCy and demonstrated how various models can differ in their performance for different entities, as shown in Table [2].*

	PER			LOC			TIME			ORG			Date		
	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R
spaCy_BERT	87	90	85	80	83	78	78	78	77	81	79	82	63	86	50
spaCy_MSA	87	89	86	82	83	80	77	73	82	63	51	82	65	73	58
spaCy_MIX	87	88	85	81	81	80	76	75	77	61	49	81	65	77	57

Table 2 how models can differ between each entity.

- *The optimal model was determined to be spaCy with the "bert-base-arabic-camelbert-msa" pre-trained model, which utilizes the Modern Standard Arabic (MSA) dataset.*

- *The evaluation results, indicate that the selected model performed well when tested on both Modern Standard Arabic and Egyptian Dialect Arabic datasets.*

- *The results shown in Figure 1 indicate that when using the best-selected model, the model is unable to predict more than 20 named entities in text written in Egyptian Dialect Arabic, despite the actual number of named entities being 23.*

Figure 1 Testing spaCy_MSA using Egyptian Dialect Arabic

- *As seen in Figure 2, when testing the model using text written in Modern Standard Arabic, the model predicts 14 named entities, whereas the actual number of named entities is 17.*

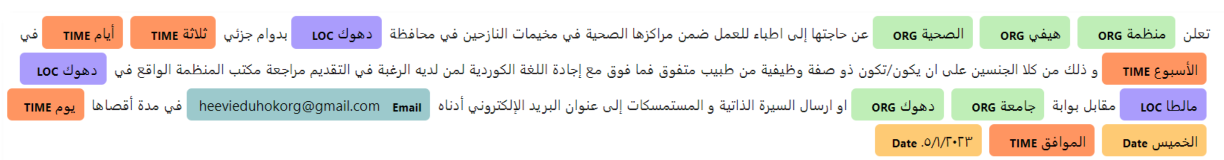


Figure 2 Testing spaCy_MSA using Egyptian Dialect Arabic

Overall, the project helped to achieve learning outcomes for the graduate program by providing hands-on experience in developing NER tools using machine learning and pre-trained models. It also helped to achieve career objectives by gaining experience in natural language processing and machine learning techniques, which are in high demand in various industries.

In retrospect, the project could have been improved by using more diverse data sets and increasing the size of the training dataset. Furthermore, it would be beneficial to use pre-trained models that were trained on more diverse data sets, which would make it more robust. Finally, it *could be considered to use additional techniques such as dictionary features which could have improved the results.*

4. Deployment Plan

In order to ensure that the Named Entity Recognition (NER) model for Arabic dialects is ready for distribution in the operational environment, a comprehensive process needs to be followed. The process should include the following steps:

a) *Conducting a final round of testing and quality assurance to ensure that the model meets the design specifications and requirements. This includes evaluating the model's performance using metrics such as precision, recall, and F1-score.*
b) *Creating documentation for the model, including user manuals, system specifications, and guidelines on how to use the model.*
c) *Developing a deployment plan to ensure that the model is installed and configured correctly in the operational environment.*
d) *Creating a monitoring plan to ensure that the model is running efficiently and effectively in the operational environment.*
e) *Conducting a user acceptance testing (UAT) to ensure that the end-users are satisfied with the model's performance and usability.*

It is important to create a smooth end-user experience, so it is vital to keep in mind the needs and preferences of the end-users throughout the process. This includes ensuring that the model is user-centered, easy to use, and providing feedback and support to the end-users.

As we are using a pre-trained NER model, we don't have to worry about maintainability issues or troubleshooting. However, it is important to keep in mind that the model's performance may be affected by the quality of the input data, so it is important to provide clear guidelines on how to prepare and preprocess the input data for the model. Additionally, it's important to ensure that the model is being used in the appropriate context and that the end-users understand its limitations and any potential biases that may be present in the model.

In order to ensure that the system or application is ready for distribution in the operational environment, it is important to conduct a thorough testing and evaluation process. This includes testing the system in a simulated operational environment, as well as performing user acceptance testing (UAT) with end-users to gather feedback and identify any issues that need to be addressed. Additionally, it is important to ensure that all necessary documentation, training materials, and support resources are available for the end-users in order for them to effectively utilize and leverage the solution.

To create a smooth end-user experience, it is essential to consider the maintainability issues and take the necessary steps to address them. This includes identifying and mitigating potential risks, as well as developing a maintenance plan that includes regular software updates, backups, and troubleshooting procedures. Additionally, it is important to establish clear communication channels with the end-users, such as a help desk or support team, to provide ongoing assistance and address any issues that may arise.

Moreover, to ensure the system or application is ready for distribution in the operational environment, it is important to document the system and its functionality, and ensure that the system's design meets the end-user's requirements. This documentation should include system *requirements, installation instructions, user manuals, and other documentation that explains how to use the system.*

Finally, it is important to have a comprehensive training plan in place for end-users, including online tutorials and webinars, as well as in-person training sessions. This will help ensure that end-users are able to understand and use the system effectively, and will help them to become more productive and efficient.

In summary, to ensure that the system or application is ready for distribution in the operational environment, it is important to conduct thorough testing and evaluations, create a smooth end-user experience, address any maintainability issues, and establish clear communication channels with the end-users. Additionally, it is important to *have comprehensive documentation and training resources available for end users.*

5. Conclusions and Future Works

Recap what you did. Highlight the big accomplishments. Finish off with a sentence or two that wraps up your project. You may explain where you think the results can lead you.
What do you think are the next steps to take? What other questions do your results raise? Do you think certain paths seem to be more promising than others? How could your solution to be improved in the future? What additional requirements you may consider enhancing the functionalities of your solution? Provide enough information as to a possible research path and why the path you took could be important and motivating. Motivation is always key in research.

6. References

7. Appendices

Include here any extra details that are not part of the main report. Try to keep the main report succinct and straight forward to read. You may have several appendices. Appendix A, Appendix B etc. and they are often good if there is detailed material you want to include in the report as a reference (e.g. detailed database / document schemas, detailed test data/results, complete set of screen shots etc>). Appendices are NOT required and will not affect your grade.