# University of Jeddah

College of Computer Science and Engineering

**Data Science**

**Report #** (Senior project)

## Analysis of Twitter data for developing a job vacancy platform in Saudi Arabia

**Instructor Dr. Abdullah Alamri**

**Students Name:**

- **Sultan Alharbi- ID: 1947457**
- **Mishal Almutiri- ID: 1948464**
- **Mohammed Almalki - ID: 1945078**

# Table of Contents

# Chapter 1

## 1.1 Introduction

The last decade can be categorized as the decade of digital transformation. Companies and governmental organizations around the world have been increasingly adopting a digital approach of managing their businesses and organizations. This includes aspects such as business processes, customer experience, warehouse management and many more. You can now study online using educational platforms, or work from home and communicate with your coworkers using Slack and Zoom while eating a meal you ordered through Hungerstation from a cloud restaurant with no physical location. The job market is no exception to this new paradigm. You are no longer expected to show up with your resume to the HR department of all the companies you are planning to apply for, instead, recruiters regularly post open positions through their websites. The recruiters of these companies usually use social media platforms to declare that they are hiring, thus, reaching potential job hunters.

Depending on the size and location of the hiring company they might either use LinkedIn or other social media platforms such as Facebook and Twitter. While LinkedIn is heavily utilized by recruiters in Europe and North America, it is yet to reach large scale adoption in the Saudi market. With its use currently limited to large and mid-sized recruiters who have a lot of human resources to keep up with this platform. Twitter, on the other hand, is the most used social media platform by Saudis with millions of monthly active users. Thousands of businesses use the platform to advertise for their products and services. More relevant to us, is the high number of job postings that are available on the website. A simple search on Twitter using any key word relevant to job postings would show you the scale and number of the open positions posted online. These job ads are from large companies and smaller ones alike. In fact, it seems that this medium is preferred by start ups and smaller businesses due its speed and the personal aspect which is important to new businesses that are just starting to build their teams.

A bottleneck that affects the practicality of using Twitter for job hunting is its unsuitable search engine. Since, the engine is universal for all kinds of tweets, it lacks some of the functionality provided by LinkedIn. More importantly is the number of spam ads that are posted by bot accounts plaguing the social media platform. When browsing any hashtag or a search result you immediately notice the amount of irrelevant tweets that poisons the twitter experience. This issue is especially damaging for those who seek job postings on the website.

While the company is actively trying to reduce the number of bot accounts spamming everywhere across Twitter. A faster and more approachable solution is needed to filter those job ad tweets. Recent developments in the Natural Language Processing (NLP) field shows promising results for sentiment analysis applications. In those applications, a Deep Learning model is trained to detect whether or not a tweet is, for example, happy or not. Such techniques could be utilized in our case to help job seekers be more efficient in their job hunt.

## 1.2 Problem Definition

Since the main issue that we are trying to solve is to classify a tweet into either spam or not, the problem could be formulated as a sentiment analysis problem which is the process of detecting positive or negative sentiment in a text. In sentiment analysis, the model is given an input sequence:

$$\boldsymbol{S} = \{w_1, w_2, \cdots, w_i, \cdots, w_n\}$$

Where $w_i$ is a word in $\boldsymbol{S}$. The task is to predict the sentiment, p̃ $\in$ $\boldsymbol{P}$, where $\boldsymbol{P}$ is the set of possible classes representing the sentiment of the sequence. In our case the sequence will be a tweet and the classes are either a real job ad or a spam tweet. To solve this problem a model $\boldsymbol{G}$ with parameters $\theta$, is to be trained on:

$$\theta = \underset{\theta}{argmin}\, \boldsymbol{L}(\mathbf{G}(\boldsymbol{S})\quad , \tilde{p})$$

Where $\boldsymbol{L}$ is an arbitrary loss function.

In addition, another deep learning model will be trained to extract important information related to the job ad. This problem is usually known as Information Retrieval.

## 1.3 Proposed Solution (The Recommended Solution )

At the mean time we are planning to use a Long Short Term Memory (LSTM) based architecture. LSTMs are widely used in sequence modeling tasks and especially in NLP. At its core, an LSTM cell is a more complex Recurrent Neural Network (RNN) cell. With its gating operations, an LSTM has longer memory and thus is better at capturing long range dependencies between the words in an input sequence. An RNN takes in a sequence of words, $\boldsymbol{S} = \{w_1, \ldots, w_T\}$, one at a time, and produces a *hidden state*, *h*, for each word. Recurrence in the name means that the RNN analyzes the input sequence by feeding in the current word $x_T$ as well as the hidden state from the previous word, $h_{t-1}$, to produce the next hidden state, $h_t$:

$$h_t = RNN(w_t, h_{t-1})$$

Once we the final hidden state, $h_T$, is reached, a fully connected layer, $f$, processes the final hidden state to produce the predicted sentiment:

$$\hat{p} = f(h_T)$$

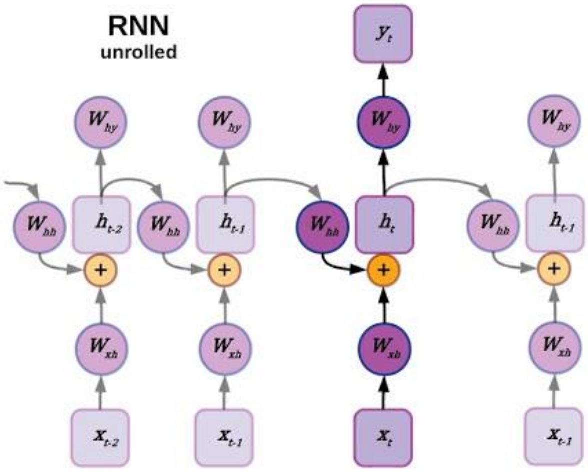


*Figure 1: An Unrolled Recurrent Neural Network*

Figure 1 demonstrates how Recurrent Neural Networks work. Notice how the hidden state of each step is fed to the next step, thus keeping an overall memory of the sequence. Although LSTMs have been widely used for sequence modeling tasks, they are no longer the State-of-The-Art, with attention-based architecture such as the Transformer dominating the NLP field. Another issue with LSTMs is their sequential nature which makes their training slow. We might go with Transformers later in the project and use the LSTMs as a baseline, but this depends on the availability of large enough datasets for training the Transformer.

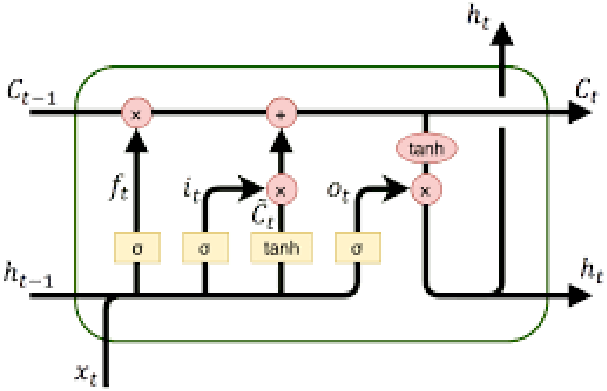


*Figure 2: Information Flow in an LSTM Cell*

Figure 2 shows the internal structure of an LSTM cell. Notice how different gates handle the flow of information. Training will determine what each gate will allow to pass and when to block certain parts of the sequence if the network believes that they would not add any valuable information.

## 1.4 Project Aims and Objectives (Project Scope)

The contribution of this project is twofold. First, we want to help ease the tiring job hunt process on people. We are currently going throw it and we understand how frustrating it can get. By doing so we are also further expanding the prospects of the Arabic Natural Language Processing field by developing datasets and models that would help Arabic NLP researchers around the world. To do this, the following points have to be achieved by the end of this project:

1- A literature review of past Arabic (and non-Arabic) Sentiment Analysis tools.
2- A Deep Learning based Arabic Tweets Sentiment Analysis model (Minimum Expected Outcome).
3- A Deep Learning based Arabic Tweets Information Extraction model (If time permits)
4- A comparison between different sequence modeling architectures, LSTMs Transformers, etc. (If time permits).

## 1.5 Target Users

Following the discussion in the previous section, we believe that the project could benefit:

1- Job seekers.
2- Recruiters (Wider reach).
3- Arabic NLP researchers.

## 1.6 Methodology

While there are many sentiments analysis and information retrieval tools and models available online, initial search showed that there are no solutions specific to job ad identification and information retrieval for Arabic Language. In addition, there is lack of datasets to support the development of such solutions. First, a dataset of Arabic tweets will be collected. Tools such as RapidMiner and Python have already been used to produce initial results. We plan to collect other metadata along with the tweets that could help in making more informed predictions such as location, date and other information related to the user posting the ad. Then each tweet in the dataset has to be labeled and categorized into either one of two classes, job related or not. The dataset will then be used to train deep learning models to achieve the above-mentioned tasks. The deep learning framework PyTorch is to be used. This Python based framework is heavily used in the literature which makes it well documented with a lot of support online. To help in reducing training time, we are currently experimenting with the use of CUDA to make use of our local GPUs for training.

## 1.7 Project Plan

For the initial plan, we are expecting that this project will have four main phases:

1- **Problem Identification and Literature Review**. In this stage, which we have already covered most of it, a problem have to be chosen and the available literature tackling this problem must be investigated. In our case, we are addressing the Arabic Sentiment Analysis for job ads. Some of the literature for the sentiment analysis problem have been read and we have now an overall idea on what we are planning to do.

2- **Dataset Collection and Preprocessing**. After covering the literature, we concluded that there are a lack of datasets tackling this issue. In this phase of the project we aim ate closing this gap.

3- **Model Training**. The initial search we have done so far gave us an idea on the type of model we are expecting to use. In this stage, we will build the appropriate model for the problem and start the training.

**4- Testing.** In this stage, excessive testing and experiments have to be carried out to ensure model performance and generalization.

## 1.8 Project Timeline (The grant chat )

| **Task** | | **Week** | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** |
| **Chapter 1** | Market Needs Identification | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | | | | | | | | | |
| | Brief Literature Review | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | | | | | | | | | |
| | Required Tools Identification | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | | | | | | | | | |
| **Chapter 2** | Detailed Literature Review | | | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | | | | | | | |
| | Survey Possible Datasets | | | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | | | | | | | |
| | Survey SOTA Tools | | | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | | | | | | | |
| **Chapter 3** | Data Collection | | | | | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | | | | |
| | Data Description | | | | | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | | | | |
| | Data Exploration | | | | | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | 🟧 | | | | |
| **Chapter 4** | Removal of Unwanted Observations | | | | | | | | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | |
| | Handling Missing Data | | | | | | | | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | |
| | Data Transformation | | | | | | | | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | |
| **Results and Discussions** | | | | | | | | | | | | ▨ | ▨ | ▨ | |
| **Final Report Submission** | | | | | | | | | | | | | 🟥 | | |
| **Presentations** | | | | | | | | | | | | | | 🟥 | 🟥 |

## 1.9 Tools and requirement

At this stage we are expecting to use the following software’s, frame works, libraries and hardware:

1- Python
   a. NumPy
   b. PyTorch
   c. Pandas
   d. Snscrape
   e. CUDA
   f. Timm
   g. scikit-learn
   h. spacy

2- RapidMinder
3- Google Colab (If CUDA did not work)

## 1.10 Conclusion

In this chapter, the Arabic job ad tweets identification problem, a specific Sentiment Analysis problem, has been formulated and the necessary background is discussed. In addition, a project plan to tackle this project is proposed outlining four major milestones to finish the project by the end of the academic year. A Gantt chart is also provided at the end of this document. We expect that this project will result in a solution to a very prominent problem covering the needs of millions of job seekers in the region.

# Chapter 2

## 2.1 Introduction

In this section, a background regarding the Sentiment Analysis problem is given in which we define the problem, the possible people, organizations and businesses needing a solution for this problem and examples on how solving this problem might help those entities. In addition a detailed literature review is carried out in which we evaluate the current research state of Arabic Sentiment Analysis.

## 2.2 Background

### Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a Natural Language Processing (NLP) task, in which the goal is to identify and quantify, subjective information. Sentiment analysis is one of the most active sub tasks in NLP [1]. The problem is essential in many fields outside of computer science such as political science, marketing and business. This is mainly due to the fact that those areas depend heavily on analyzing, understanding and engineering opinions. This application has witnessed an increase in popularity in the last few years since humans are giving their thoughts and opinions more freely than what they are used to before. This is driven by the ease and anonymity social media platforms are providing to individuals. In addition, the large amounts of data users post online on social media platforms are empowering more robust models that organizations can trust [2]. Sentiment analysis has become a must have tool to understand users' intention. It allows organizations to save time and effort by automating the feedback analysis process using responses from surveys and online interactions. Which in turn provides businesses and governmental organizations with the ability to know what causes their customers satisfaction or distress. Thus, allowing them to optimize their products and offerings to the needs of their customers.

For instance, you may learn why consumers are satisfied or dissatisfied at each point of the customer journey, by automatically analyzing thousands of open-ended comments from your customer satisfaction surveys using sentiment analysis. In addition, you could monitor brand sentiment so that you can identify and address displeased customers as soon as possible. To determine whether you need to change and modify parts of your process, you might wish to compare sentiments from two different quarters. Then you may delve more deeply into your qualitative data to discover the causes of sentiment changes.

There are multiple ways in which the problem of sentiment analysis is posed including:

- Graded Sentiment Analysis

In this type, textual snippets of data are given a polarity depending on how positive or negative the sentiment they convey. The polarities might range from Very positive to Very negative. This is usually used to interpret 5-star ratings in a review. Hence, Very Positive = 5 stars and Very Negative = 1 star.

- Emotion Detection

Emotion Detection used to utilize lexicons which are key word lists and the emotions they convey. Lexicons are rigid, and they do not take the context around the keywords into consideration. People express emotions differently. Words that typically express anger, like bad or kill (Example: your customer support is killing me) might also express happiness (Example: you are killing it). With recent advances in machine learning, data driven approaches are more common now. Machine learning based methods are robust and many of them can capture the context very well.

## 2.3 Related Work

Most of the existing research about fake job ad detection focuses on English content. In fact, there is a limited number of research projects that have been carried out on Arabic content in social networks. We will discuss some of them as follows:

1. According to a study by Ibrahim Nasser and Amjad Alzaanin et al. (2021), titled “Machine Learning and Job Posting Classification: A Comparative Study” [3]. They used a dataset from Kaggle which was named “Real and Fake Jobs data”. They investigated multiple machine learning classifiers such as Multinomial Naive Bayes, Support Vector Machine, Decision Tree, K Nearest Neighbors, and Random Forest to be able to classify the job ads. They cleaned and pre-processed the data, then applied TF-IDF for feature extraction, and finally applied the different classifiers to the data.

2. In a study by Shawni Dutta and Samir Kumar Bandyopadhyay et al. (2020), titled “Fake Job Recruitment Detection Using Machine Learning Approach” [4]. They proposed an automated tool to overcome fraudulent job posts over the internet using machine learning-based classifiers. Again, they used the same data from Kaggle as the first research which is “Real and Fake Jobs data”. They used both single classifiers such as Naïve Bayes, Decision Tree, and K Nearest Neighbors classifiers. for ensemble classifiers, they used the AdaBoost and the Gradient boosting algorithms.
3. In another study by Gasim Alandjani et al. (2022), titled “ONLINE FAKE JOB ADVERTISEMENT RECOGNITION AND CLASSIFICATION USING MACHINE LEARNING” [5]. His aim was to reduce the number of fake ads using machine learning. He used many classifiers to figure out which one managed to achieve the best performance. He used the same open-source data from Kaggle. This research focuses on applying SVM, Multinomial NB, Decision Tree, Random Forest, and K-nearest neighbor classifiers.

4. In a study by Aaroh Baweja and Pankaj Garg et al. (2019) titled” Sentimental Analysis of Twitter Data for Job Opportunities” [6]. They tried to use the concept of sentiment analysis by applying it to social media platforms, especially Twitter. In addition to applying the machine learning model to secure the position openings. They used a dataset from Twitter and then applied a sequence of data preprocessing to clean the data.

5. In a study by Lekha R. Nair and Sujala D. Shetty et al. (2015), titled “STREAMING TWITTER DATA ANALYSIS USING SPARK FOR EFFECTIVE JOB SEARCH” [7]. They tried to address the issue of real-time big data and filtering the real job opportunities out of the millions that were posted. Their aim was to classify and categorize these jobs to make the search easier. They analyzed very large-scale data using Apache Spark.

6. In a study by Sokratis Vidros, Constantinos Kolias, Georgios Kambourakis, and Leman Akoglu et al. (2017), titled “Automatic Detection of Online Recruitment Frauds: Characteristics, Methods, and a Public Dataset” [8]. In this research, they contribute in two areas which are investigating the diverse aspects of Online recruitment Fraud especially employment scam and they offer the first available dataset which contains real-life legitimate and fraudulent job ads that were published in English. They applied different machine learning models such as Logistic Regression, Random Forest, and Decision Tree.

7. In a study by Bandar Alghamdi and Fahad Alharby et al. (2019), titled “An Intelligent Model for Online Recruitment Fraud Detection” [9]. In this research, they tried to reduce the loss of money for individuals and corporations by creating a model that has the ability to detect fraud in online recruitment posts. They used a free dataset called Employment Scam Aegean Dataset (EMSCAD). For the modeling, they used an ensemble model which is the Random Forest.

8. In a study by Aashir Amaar, Wajdi Aljedaani, Furqan Rustam, Saleem Ullah, Vaibhav Rupapara, and Stephanie Ludi et al. (2022), titled “Detection of Fake Job Postings by Utilizing Machine Learning and Natural Language Processing Approaches” [10]. They proposed a methodology that combines the concepts of natural language processing and supervised machine learning techniques to detect fraudulent job ads from the internet. They used two methods for feature extraction which are Term Frequency-Inverse Document Frequency (TF-IDF) and Bag-of-Words. In addition to using six different machine-learning algorithms. They faced an imbalance data problem but they overcame it using the adaptive synthetic sampling approach (ADASYN).

9. In a study by C Prashanth, Deepanjali Chandrasekaran, Bhuvanashree Pandian, Kavitha Duraipandian, Thomas Chen, and Mithileysh Sathiyanarayanan et al. (2022), titled “Reveal: Online Fake Job Advert Detection Application using Machine Learning” [11]. They developed a machine learning web application to identify fake online job advertisements. They called the application Reveal.

10. The last study by Tao Jiang, Jian Ping Li, Amin Ul Haq, Abdus Saboor, and Amjad Ali et al (2021), titled “A Novel Stacking Approach for Accurate Detection of Fake News” [12]. They used two datasets which are ISOT and KDnugget datasets. Also, they used the term frequency-inverse document frequency for word embedding. In addition to using different machine learning algorithms adding a novel of stacking models together helped them to achieve high accuracy.

## 2.4 A Comparison between proposed system and literatures

As mentioned above, most researchers developed systems that are based on different machine learning algorithms such as Logistic Regression, Support Vector Machine, and others. These systems mainly focused on detecting fake job ads that were published in English. To propose the systems and train the models, they used different datasets such as Real and Fake Jobs data which is a free dataset from Kaggle, Employment Scam Aegean Dataset (EMSCAD), ISOT dataset, and KDnugget dataset.

Our contributions will be to develop a job ads classification system using Long Short-Term Memory (LSTM) which is a deep learning model. Also, we will focus on collecting the dataset from Twitter, especially the jobs that were posted in Arabic

## 2.5 Conclusion

In this section we have carried out a literature review to assess the current state of Arabic Sentiment Analysis research. We have found out that there is a lack of works covering the Arabic job ads tweets identification despite its significant importance for job seekers.

# References

[1]

Deep Learning for Sentiment Analysis: A Survey (arxiv.org)

[2]

An Empirical Analysis of Attitudinal and Behavioral Reactions Toward the Abandonment of Unprofitable Customer Relationships: Journal of Relationship Marketing: Vol 9, No 4 (tandfonline.com)

[3]
https://www.researchgate.net/publication/344406673_Machine_Learning_and_Job_Posting_Classification_A_Comparative_Study

[4]
https://www.researchgate.net/publication/341325717_Fake_Job_Recruitment_Detection_Using_Machine_Learning_Approach

[5]

https://www.researchgate.net/publication/359700215_ONLINE_FAKE_JOB_ADVERTISEMENT_RECOGNITION_AND_CLASSIFICATION_USING_MACHINE_LEARNING

[6]

https://www.irjet.net/archives/V6/i11/IRJET-V6I11283.pdf

[7]

http://www.jatit.org/volumes/Vol80No2/17Vol80No2.pdf

[8]

https://www.mdpi.com/1999-5903/9/1/6

[9]

https://www.researchgate.net/publication/334365773_An_Intelligent_Model_for_Online_Recruitment_Fraud_Detection

[10]

https://link.springer.com/article/10.1007/s11063-021-10727-z

[11]

https://ieeexplore.ieee.org/document/9752784

[12]

https://ieeexplore.ieee.org/document/9343823

# Chapter 3

## 3.1 Introduction

In this chapter we discuss the process we followed to collect the dataset we plan to use in our project. Then, we describe in details the characteristics and properties of our dataset and the logic behind the decisions we made when collecting the data samples. Finally, we present some exploratory data analysis to better understand the dataset.

## 3.2 Data Collection

For data generation, the Python library Snscrape was used to scrape Twitter for relevant Tweets. Before going with this option, we considered using RapidMiner tool for Twitter scraping. The issue with this tool is that it limits the number of scraped tweets to 15K every 15 minutes. In addition, the tool only scrapes tweets from the last two weeks. This might significantly degrade the dataset's quality since collecting thousands of tweets in such a limited time period would mean many of those tweets would repeated. Hence, to enhance our dataset quality and to ensure that the model would be trained on a diverse samples and thus generalize, we decided to opt for using the snscrape Python library which does not suffer from those issues.

Different queries have been used. Table 1 shows the used queries. We have decided to use this list after conducting a brief search on twitter and collecting the phrases that frequently occurs in Twitter job ads. We also asked our friends and family members who recently went through the job-hunting process to give us relevant key words to use. For each query, a large number of Tweets are collected for processing. The processing and preparation details will be outlined in the Preparation section of the report.

**Table 1: List of used queries when generating the dataset**

| |
|---|
| شواغر |
| وظائف |
| تعلن عن حاجتها |
| فتح باب التقديم |
| وظائف شاغرة |

For Each query, the data collected, numbered in the thousands, is processed using the following pipeline:

1- Duplicate Tweets are removed in order to not bias the model.
2- Tweets with zero Likes, Retweets and Replies are removed because after looking into the dataset, such tweets are usually bot generated. Since these tweets are the majority, they might bias the model, thus we remove large part of them. We keep some of them so that the model would learn to recognize them.
3- Each tweet is then manually annotated to check whether it is a job-related tweet or not.

## 3.3 Data Description

To ensure model generalization we made sure during the generation process that the dataset contains tweets from different Twitter users, and spanning different years and months within the years, thus having different ways of framing the job ad and different types of jobs. The latter is emphasized here since there are some seasonal jobs that could bias the dataset if we focused on a narrow timeline. Such jobs include those that are needed during the Hajj or Riyadh and other cities seasons.

In addition, while conducting the initial search process, we noticed that there are large number of tweets that discuss topics relevant to jobs and the job market in general. Such tweets might be posing certain statistics about the job market such as the number of new jobs created in a particular year, or unemployment rates and so on. Other tweets are basically normal people complaining about the status of the job market or about being unemployed or are just unsatisfied with the recruitment process. These tweets are extremely common and hence it was essential to include them in the dataset to ensure models trained on our dataset would not be tricked into categorizing such tweets as job ad tweets.

Other category that is not as common as the previous one, yet it might cause some issues for our model, are tweets that declare the start of the application and admission process for university programs. Such tweets share key phrases with job ad tweets and thus, we added a sample of them to our dataset. Examples of the discussed categories are outlined in table 3. Notice how the words written in red are words expected to occur in job ads tweets. If the dataset set does not take this into account, that is, if we only collect job ad tweets as Class 1 tweets and irrelevant tweets as Class 0 tweets, the model will probably learn to look for those key phrases and assign tweets having them as Class 1. By adding those negative examples, we are forcing models to be trained on our dataset to look for other defining patterns of job ad tweets.

Finally, we added tweets that do not contain anything related to job ad tweets and with misleading key words or phrases. These tweets along with the previous two categories are labeled as non-Job Ad tweets, or Class 0. While tweets posting job ads are Class 1. Figure 1 shows examples of different tweets that cover the categories described above. Table 2 gives the different attributes we collected along with the tweets.

**Table 2: List of dataset attributes**

| **Attribute** | **Explanation** |
|---|---|
| URL | Tweet link |
| Datetime | The date and time of the tweet |
| Text | The tweet text |
| Username | The username who tweeted the tweet |
| Likes | Number of likes the tweet got |
| Quote-tweets | Number of quote tweets the tweet got |
| Cleaned | Text after preprocessing |
| Class | 0 or 1 |
| Replies | Number of replies the tweet got |
| Country | The country from which the user is tweeting |
| Confusing | Yes: Confusing - No: non-confusing |
| Retweets | Number of quote tweets the tweet got |

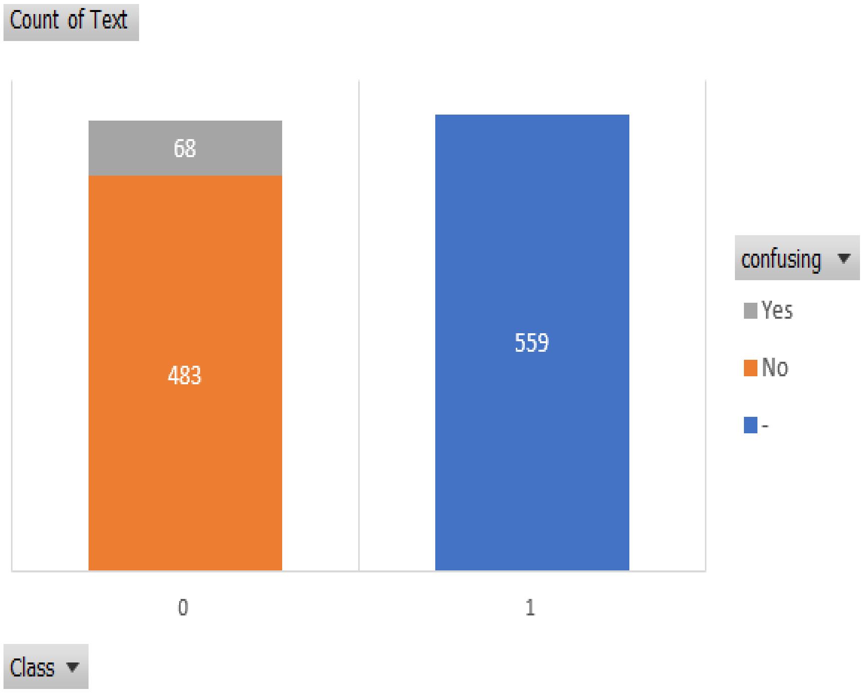
**Table 3: Possible confusing tweets. The keywords shared with job ad tweets are highlighted in red.**

| Tweet | Comment |
|---|---|
| ي جماعة اللي يعرف وظائف لتخصص نظم معلومات ادارية يتواصل معايا الله يسعدكم 💐 | A tweet of a person looking for jobs. |
| #ازمه_عطاله_البكالوريوس لا تخصصات جديدة متاحة ولا فيه وظايف للتخصصات الي كل الخلق داخلها وفيها تكدس خريجين عاطلين. | A tweet from a person expressing frustration with the job market. |
| فتح باب التقديم على #برامج_الدراسات_العليا المهنية والتنفيذية ب #جامعة_جدة<br>للاطلاع على تفاصيل البرامج المهنية والتنفيذية وشروط القبول فيها: https://t.co/PZybefVOfS https://t.co/wlAs1Fwtrg | A university announcing the start of receiving applications of their Graduate Programs. |

## 3.4 Exploratory Data Analysis

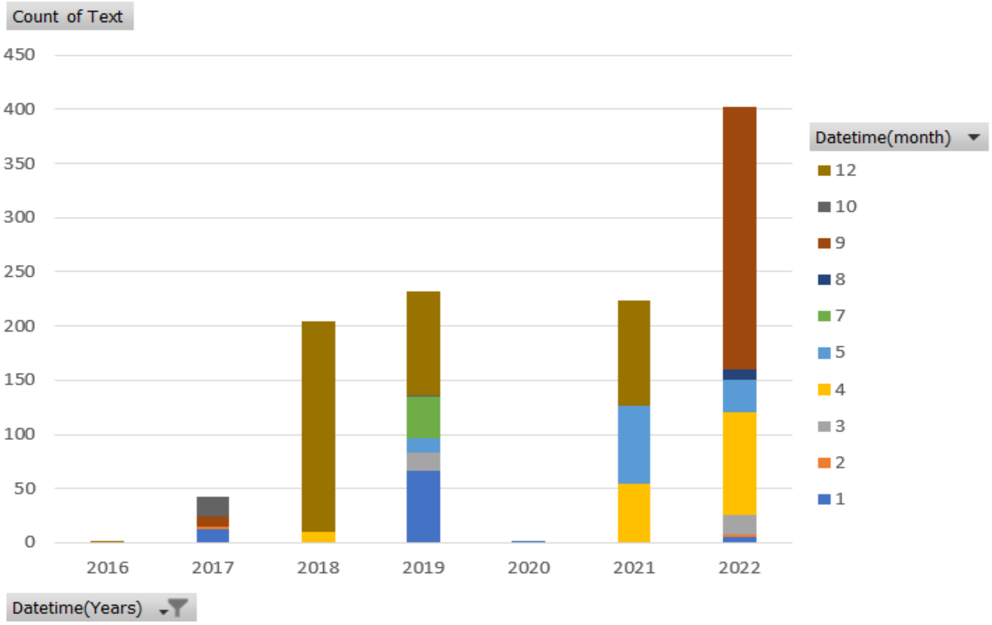
The dataset contains a total of 1110 Tweets distributed across the classes described earlier. Figure 1 shows the tweets counts by class. There are 559 job ad tweets in the dataset. The other class is for tweets that are not job ad tweets. The dataset contains 551 of this class distributed. between two subcategories as discussed in the data description section. 483 are confusing non-job ad tweets which contain keywords and phrases that usually occur in job ad tweets. The other subcategory, represented by 68 tweets is a collection of normal, irrelevant tweets whose purpose is to provide the model with a template for general non-job ad tweets.

**Figure 3: Number of Tweets by Class**



Another holistic thing we want to investigate in our data is the dates at which the tweets were published. We look here into both the year and the month of publication. We are focusing only on the job ad tweets in our dataset. Non job ad tweets are not considered for this analysis as the date of publication is irrelevant. The importance of this analysis is that the MENA job market has witnessed a dramatic change in the last few years. This change is driven by Covid-19 and more importantly to the KSA case is Vision 2030. These recent events caused more openness towards work from home and modernized marketing, HR and the process followed by job recruiters. As shown in Figure 2, we made sure when collecting our dataset to focus on tweets from recent years with varying distributions across months within each year.

**Figure 2: Tweet density/year in the generated dataset**

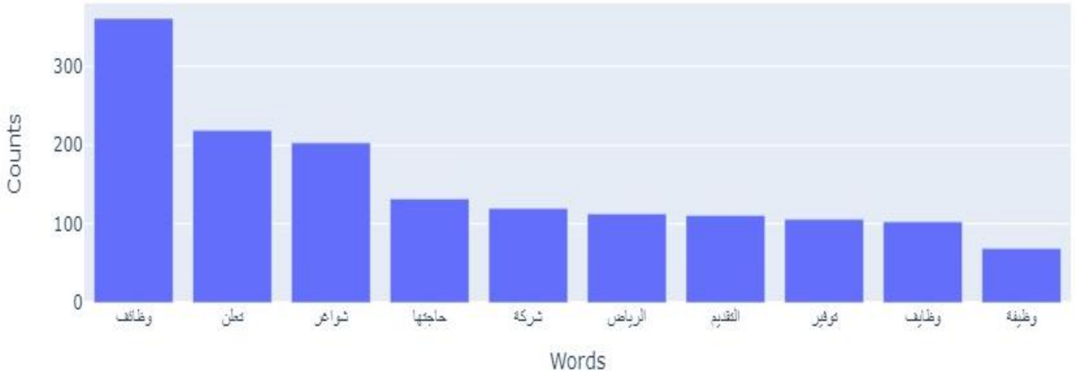


Although our dataset is a bit more focused on 2019 data, going through samples from the last three years shows similarity between job ad tweets thus it is safe to go with the current distribution.

Moving to a per tweet analysis of word frequencies would help us in capturing new keywords that could be used to expand the dataset. To do this first we have to further process the text in each tweet and use those tweets to generate a vocabulary from which we can gain insights on word frequencies in the dataset. First, we remove English words from each tweet. Although English words would play an important role when training the model, for the purpose of this analysis we remove non-Arabic words and symbols as they are not of interest at this stage. Then, punctuations and diacritics are also removed along. Finally, to normalize the words, we remove repeated letters as some people use this type of writing for emphasis. After this processing pipeline, we split each tweet into words and add them to global list or a vocabulary. We also keep track of the frequency of each word.

Figure 3 shows the most repeated words in the vocabulary. Note that words used in queries to generate the dataset are discarded (Words in Table 1). There are two main takeaways from this graph. First, we can see that Riyadh region is the most frequent location for which there are Twitter job ads and a Bachler degree is the most looked for degree in those ads. The other takeaway, one which could be an inspiration for additional features in our system, is the use of keywords such as: "خبرة", "شركة", and "الرابط". The presence of these words calls for another model whose job is to recognize entities and retrieve important information. These words show that the dataset might be ready for such tasks when annotated properly.

**Figure 3: Word frequency in the vocabulary**



## 3.5 Conclusion

This chapter explains the process followed to collect the dataset. The characteristics of the dataset and the logic behind the decisions we made when collecting it are outlined. Finally, some exploratory data analysis to better understand the dataset is carried out.

# Chapter 4

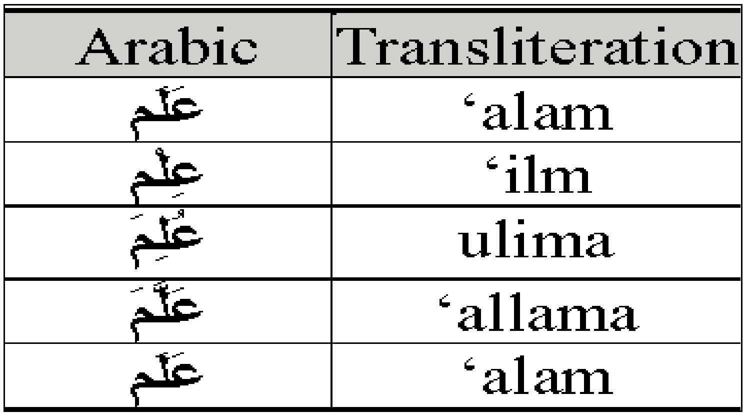
## 4.1 Introduction

In this chapter we discuss the problems that are present in the dataset we have collected. These problems are usually a characteristic of all Natural Language Processing data and follow a specific pipeline to solve and prepare for downstream training. There also a couple of problems specific to Arabic NLP that will be discussed and tackled in this section.

## 4.2 Problems with the Dataset

Due to the fact that different people write differently, datasets prepared for NLP tasks usually suffer from extreme variations that could mislead models trained on them. This is especially the case with the Arabic language. People from different parts of Saudi Arabia, for example, would write a sentence that conveys the same information but using different words or the same words written differently. This is issue is magnified with people not writing in a standard way but rather carrying on their verbal dialects when writing. Another major issue is that the same word could mean different things depending on its location and context. As can be seen in Table 1, the word “علم” could mean multiple things and the meaning could be conveyed by using diacritics. The issue with diacritics is that they are usually used by many in Saudi Arabia in a non-standard way where they use it not for making the sentence more clear but rather in a random way and thus they need to be removed.

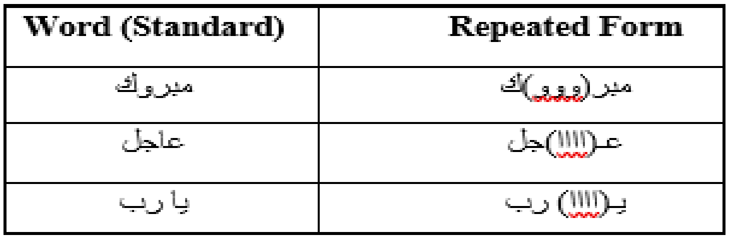
**Table 4: Diacritics effect on word meaning**



| Arabic | Transliteration |
|---|---|
| عَلَم | ‘alam |
| عِلْم | ‘ilm |
| عُلِمَ | ulima |
| عَلَّمَ | ‘allama |
| عَلَم | ‘alam |

Another issue is the use of repeated words when writing. This is usually done by Saudis to convey urgency, excitement or happiness as shown in Table 2.

**Table 5: Repeated letter in Arabic Writing**



| Word (Standard) | Repeated Form |
|---|---|
| مبروك | مبر(ووو)ك |
| عاجل | عـ(اااا)جل |
| يا رب | يـ(اااا) رب |

In addition, the tweets in the dataset had a major discrepancy that could cause model divergence if left unsolved. The issue was the use of punctuation, which is a problem that is not specific to Arabic NLP as other languages suffer from too, unlike the previous two issues. Not all people use punctuation, some do and some don’t. Even those who do use them, they don’t use them in a standard way and each would add their own personal style to them. Refer to Table 3 to see examples of this phenomenon.

**Table 6: Nonstandard use of punctuation**

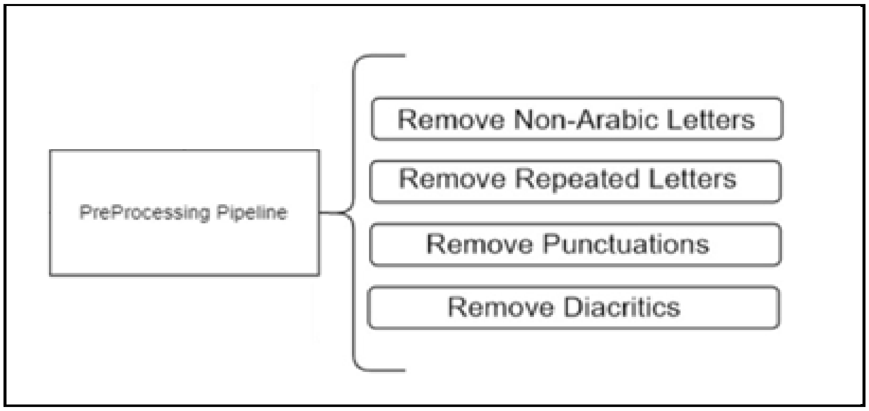
| Tweet | Comments |
|---|---|
| والله شي يقهرر نأخذ الشهادة وما نلقى وظايف الله المستعان."<br>💔🧎‍♀️ #ازمه_عطاله_البكالوريوس | In this tweet, the author used the quotation mark in a non-standard way |
| دحين عندي سؤال بالأصح اسئله..<br>انا ترم وادخل جامعه وافكر اتخصص وراثه طبيه  سؤالي<br>التخصص حلو ولا مو حلو ؟؟ + له وظائف يعني اش اقدر اشتغل<br>واذا درستو يعتبر دكتوره جد؟؟#كلمه_اخيره_٢٠١٨ | In this tweet, the author used double full stops to separate sentences |
| @s9nfcاهم شي نبي وظايف وكثر اللع خيرهم:) | This user is using single parathesis with a colon as a smile emotion |
| اللهم اجبر خاطري جبراً انت وليه. | Normal use of a full stop |
| شواغر وظيفية نسائية في جازان<br>مطلوب تخصص صحافة و إعلام<br>الشروط و المهارات طلاقة في اللغة الانجليزية<br>التقديم يرجي ارسال السيره الذاتيه<br>عبر الايميل<br>tele2030jpc@gmail.com | In this example, the author does not use punctuations whatsoever |

The final issue with the collected dataset was the presence of a lot of non-Arabic words. These come mainly from user handles that people mention when writing or replying to a tweet. Another source of non-Arabic words is website links that job advertisers put in their job ad tweets. Those have to be removed because the word embedding backbone we are planning to use was trained on purely Arabic data. There are some models trained on Arabic + English that we might consider later.

## 4.3 Data Preparation and Preprocessing

To address these issues the pipeline shown in figure 1 is designed and followed. The implementation was done using Python. First, the data is loaded from the CSV file using Python’s Pandas library. Then using Python’s string processing capabilities we were able to write a code that operates on the data on tweet at a time. Then, all non-Arabic strings are removed from each tweet and the tweet is fed to the next stage of the pipeline. In the next stage repeated letters are removed.

**Figure 4: Preprocessing Pipeline**

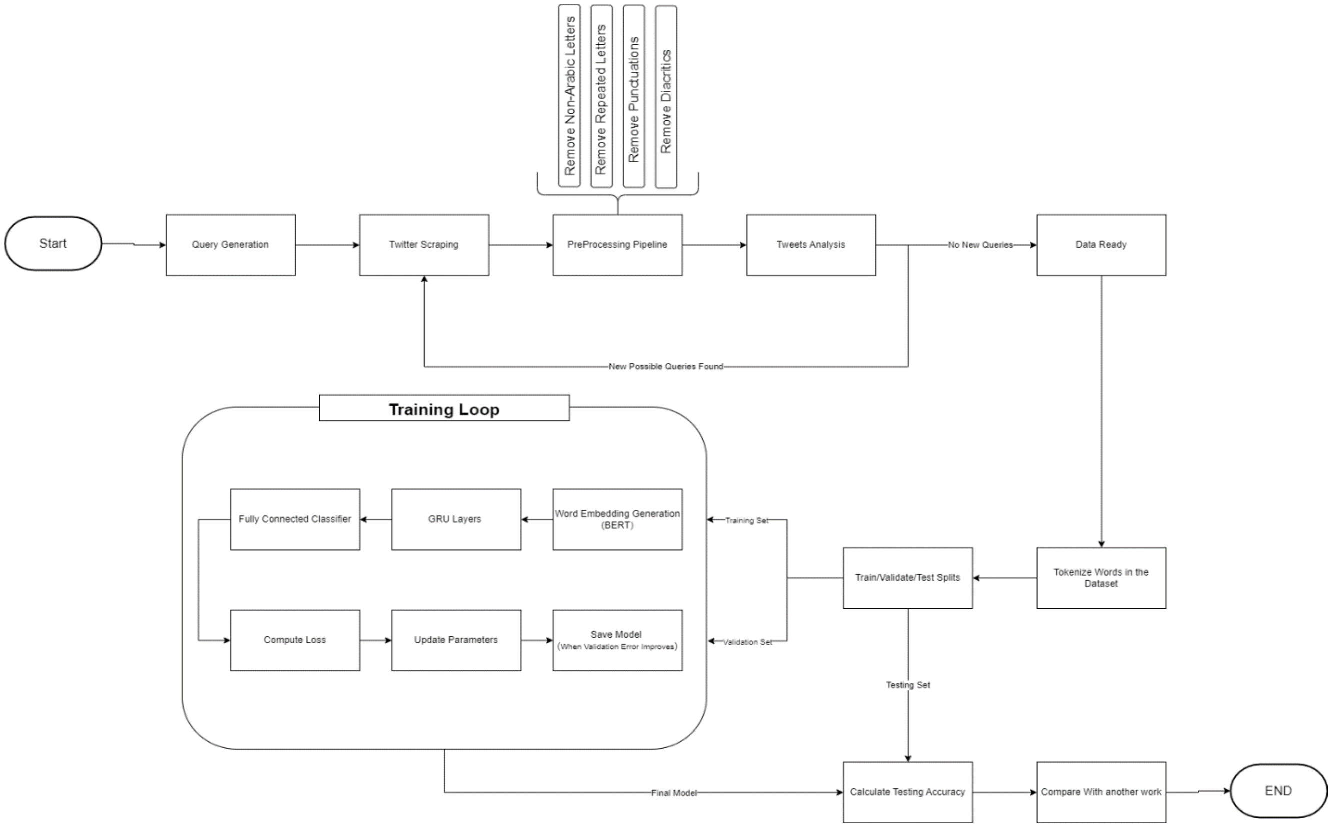


In this stage another important process is carried out in which letters that have multiple variations are normalized, that is all casted to on of those variations. Such letters include: أ, ا, إ, آ this same process goes on other similar letters that have letters that have "همزات". After that, the tweet is fed into the last two components of the pipeline which are the removing punctuations and diacritics part. These parts’ names are self-explanatory and they do just as the name suggests. Some words have repeated letters naturally, they however rarely occur and reducing them would not do much harm and from reviewing the literature it seem that normalizing them as will is a normal practice so we followed them.

## 4.5 Conclusion

This chapter discussed the problems that occur in NLP datasets. We focused more on Arabic NLP challenges as our project is focusing on the Arabic language. We finally outlined the pipeline we designed and followed to solve these issues.

## Model Architecture (with clarify what will happen in the next semester)



This model show what will be happen in next semester first We will Do Query Generation After that will move to next stage that will do twitter scraping based in out query after that will do preprocessing (Remove punctuation , Repeated Letter , Remove non Arabic letter , Remove Didactics ) After that we will check the data if there need to do new query to get data will back to twitter scrapping Stage and Collect data again with New query after that Will Do Preprocessing Again And We will check the data again  after that the data will be ready to use in next Semester.

**What Will Happen in next semester in general:**

First, we will do tokenize the words in our dataset (one hot encoding) after that we will split the data into Train/Validate and test after that the train/validate will enter to training loop.

In training loop first we will use already pretrained Word Embedding Generation (BERT)  for Arabic Will Be Already Pretrained after that in training the words in dataset will Take in Sequential using one of GRU Layers take it word by word and after each word in this stage will get an hidden state so last word will contain hides state for all sentence after that will enter to Fully Connected Classifier  for Classification after that usually in train/validate for each iteration will do Computer loss and based in loss the Parameter Will be Updated in Update parameter stage until get A good result for training loop .

After Training loop will do calculate accuracy in calculating accuracy stage by using test data and the accuracy will calculating with confusion matrix.

# Chapter 5

## 5.1 Introduction

In this chapter, we will discuss the various tools and frameworks that were used for the modeling-building phase. We used Long Short-Term Memory (LSTM) to be our main model and we will discuss it in detail. Also, we will mention the parameters that were used in LSTM and the selection criteria.

## 5.2 Experiments setup and tools

To build a powerful deep learning model such as LSTM, we had only two choices, the first one is to build it from scratch which means building all model functions containing layers configuration, forward propagation, backward propagation, training process, and so on. Or the second choice is to use a framework that contains various deep learning models such as Recurrent Neural Networks, Convolutional Neural Networks, and many others.

There are various deep learning frameworks such as TensorFlow, PyTorch, Keras, and others. In our implementation, we used TensorFlow and Keras.

TensorFlow is an end-to-end open-source framework that is used for artificial intelligence and deep learning. It was developed by Google. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML, and developers easily build and deploy ML-powered applications. In addition to that, TensorFlow is usable and can be customized based on your target.
TensorFlow has a high-level API called Keras which is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

In addition to the use of TensorFlow and Keras in model building, we used Sklearn for data preparation to be formatted as the model expects. Scikit is one of the offices for machine learning specializing in Python. Its contents include algorithms and methods used in the field of prediction, in addition to its use in the stage of data filtering and the evaluation of prototypes.

## 5.3 Initial parameters and selection criteria

### 5.3.1 Data preparation for training the deep learning models

It has been demonstrated that data preparation takes up 80–90% of machine learning development time. Because of this, even the finest machine learning algorithms will not function effectively if they are not trained on high-quality datasets that are free of biases or errors. Preparation is therefore essential before entering data into a model, such that the accuracy of the data is ensured, which in turn leads to correct insights.

#### Tokenization:

This is a way to break up a long string of text into smaller chunks called tokens. Tokenization reduces long texts like sentences to their simplest constituent words, and token borders are also present.

#### Pad-sequences:

The deep learning models use equal-length inputs. Therefore, inputs should be padded to an equal length before training the model. For the most part, this is done by adding 0 at the beginning of each sequence until it reaches a length equal to the longest sequence or a certain common length which is known as ‘zero padding’.

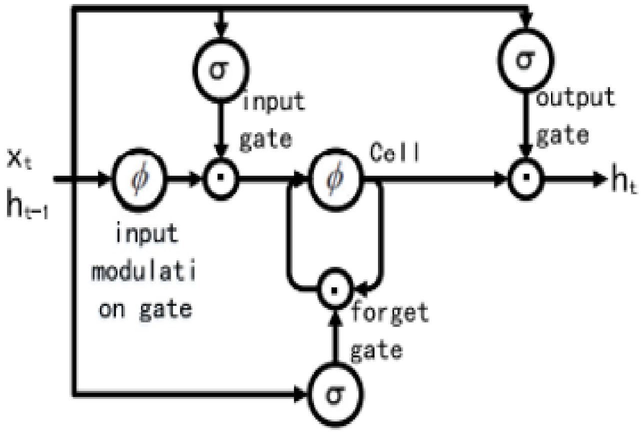
#### Embedding:

High-dimensional vectors can be translated into a low-dimensional space through a process called ‘embedding’. Using this process enables big data inputs to be handled by the deep learning model. However, with the advent of pre-trained models like BERT, embedding approaches have advanced even further. So, Word2Vec- or GloVe-based embedding layers, which only offer a single independent context representation for each token, BERT uses the complete sentence to construct token-level representations.

### 5.3.2 Model Building:

Recurrent neural networks (RNNs), such as LSTM, are commonly employed in deep learning and can analyze long-term data sequences. LSTM has become a popular strategy for forecasting time series because of its ability to learn long-term data sequences. A long-term forecasting strategy based on the benefits of LSTM is proposed to anticipate time series with strong periodicity. LSTM neural networks have been found to be powerful in handling data streams. So, analyzing sequential data requires the usage of the LSTM RNN methodology. For example, in stock forecasting, text generation, and voice recognition, recurrent neural networks are commonly used to anticipate future information based on stored data. In addition, the pace at which information is sent to the cell is controlled by an LSTM gate structure that was specifically built for this purpose. To reflect the number of times each part has been passed through, the sigmoid layer produces an integer between 0 and 1.

#### LSTM Architecture



The above figure demonstrates that the cell, input gate, output gate and forget gate are all components of a typical LSTM unit. Using three gates to govern data flow, the cell may store values for any number of time periods. It is regulated by a number of gates that determine whether or not the cell will maintain or reset the value. Additionally, there is an input modulation gate called ct, which is used to control whether the current cell value (forget gate ft) should be forgotten, inputted (input gate it), and then outputted (output gate ot).

The model consists of a group of layers including the embedding layer, two LSTM layers, one dropout layer, and finally two dense layers. This architecture was found to be the best one after conducting several trials. Finally, the model was compiled.

The embedding layer: This is responsible for converting each word into a vector with a defined size, the output dimensions were defined to be 500 and the input length was set to 60.

The first LSTM layer: The number of neurons was defined to be 128, this number represented the number of hidden states in this layer. The return_sequences were set to be true, as the layers of the LSTM were stacked, and all the hidden states of each time step needed to be output.

The dropout layer: This is used to prevent the model from overfitting, a common problem in deep learning, especially in the case of training deep learning models using a small dataset. The ratio was set to be 0.50, which meant that 50% of the network neurons would be frozen in each iteration.

The second LSTM layer: This was added, but this time with only 64 neurons.

The first dense layer consists of 10 neurons acting as a fully connected neural network.

The activation function was set to be tanh for both the LSTM layers and the dense layer.

The final layer: This was the dense layer that was responsible for predicting the output. It had 1 neuron and a sigmoid as the activation function. The activation function adds more nonlinearity to the network which helps to deal with more complex problems.

Although there are several types of activation, the sigmoid was selected as the sigmoid function guarantees that the output will always be between 0 and 1.

### 5.3.3 Training parameters

The model was compiled by setting the optimizer, loss function, and matrix, the model was trained by using the fit method that takes some hyperparameters such as verbose parameter, percentage of validation data, and the number of epochs which the model was trained for a different number of epochs to compare the results and find out the best performance.

The model was compiled by setting the optimizer, loss function, and matrix, the model was trained by using the fit method that takes some hyperparameters such as verbose parameter, percentage of validation data, and the number of epochs which the model was trained for a different number of epochs to compare the results and find out the best performance.

Optimizer: The Adam optimizer, a combination of the Adagrad and RMSProp algorithms, was used to help the model converge. In general, the optimizer is used to adjust the network parameters to reduce the loss.

Loss function: binary cross entropy was used as the most valuable type of loss function with a binary classification task.

Matrix:

The accuracy was used in several trials to calculate the matrix which represents the ratio between true samples and all the other samples.

## 5.4 Conclusion

In this chapter, we illustrated our methodology in detail starting with the data preparation steps including tokenization and pad sequence until the stage of model building. LSTM was chosen to be our main model, the model architecture consists of an embedding layer, 2 LSTM layers, a dropout layer, and finally 2 dense layers. furthermore, the model was compiled with Adam optimizer, binary cross entropy as loss, and accuracy to be the metric. Finally, the model was trained for many epochs with a validation size equal to 0.1.

# Chapter 6

## 6.1 Introduction

In this chapter, we will present the final step which was to test the LSTM model performance. the performance was measured on testing data representing 20% of the overall dataset size. Different evaluation metrics were used such as accuracy, precision, recall, and f1 score. Finally, we will discuss the model results.

## 6.2 Performance Evaluation Metrics

The evaluation metrics are used to measure how good the model performance is, we will start with the base one which is the confusion matrix.

### 6.2.1 Confusion matrix

A tabular representation is frequently used to describe the performance of a Machine Learning classifier in predicting unseen data when the actual values are known. Also, it can clearly explain the classifier's ability to separate one class from the other in binary classification problems.

| **Actual label** | **Predicted label** | |
|---|---|---|
| | **1** | **0** |
| **1** | True Positive | False Negative |
| **0** | False Positive | True Negative |

### 6.2.2 Accuracy

is considered the simplest metric to implement. It indicates the number of correctly predicted ads, either Related to ad or not. Moreover, it can be calculated from the number of correct predictions to the total number of predictions as shown in the equation.

$$\text{Accuracy} = \frac{TP\ +\ TN}{TP + TN\ +\ FP\ +\ FN}$$

### 6.2.3 Precision

It measures the model's ability to distinguish the positive samples. The precision metric determines the proportion of positive predictions that were actually correct. It can be calculated as the ratio of the number of True Positive predictions, which are the not relate to ads ads that are correctly classified to belong to the not relate to ads class, to the total number of positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP}$$

### 6.2.4 Recall

It evaluates the model's ability to classify the positive samples (Sensitivity or True Positive rate). The recall metric determines the proportion of the actual positive samples that were identified incorrectly. It can be calculated as the correctly predicted positive samples to all the samples with actual label positive.

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN}$$

### 6.2.5 F1 Score

is a widely adopted metric to evaluate the binary classification model performance based on how well it predicts the positive class. It is a combination of precision and recall. The harmonic mean of both can be used to calculate the F-measure by the following formula.

$$\text{F-measure} = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

## 6.3 Experiments Results

In deep learning, an experiment is a process of applying specific model to dataset to find out how good model’s performance is. In our case, we tested LSTM model on all the following:

- Training data: it contains 72% of total dataset.
- Validation data: it contains 8% of total dataset.
- Testing data: it contains 20% of total dataset.

Our LSTM model managed to achieve 0.987 accuracy on the training data, 0.955 on the validation data which are acceptable results.

Furthermore, we tested the model performance on the testing data but not only using the accuracy metric, using accuracy, precision, recall, f1 score.

LSTM managed to achieve an acceptable result on testing data as follows:

| **Metric** | **Result** |
|---|---|
| Accuracy | 0.9663 |
| Precision | 0.9661 |
| Recall | 0.9664 |
| F1 Score | 0.9663 |

## 6.4 Discussion

As we can see in the previous table, the LSTM accuracy decreased on the testing set compared to training and validation sets which was expected as testing data is unseen data. Taking the slight difference into our consideration, the model performance can be considered acceptable.

Also, it is clear that all evaluation metrics results are above 96 which is a good sign that our LSTM has trained well and is expected to perform well on future job ads and hopefully contribute to saving people from not relate to  job ads.

## 6.5 Conclusion

In this chapter, we were trying to evaluate the LSTM performance according to the different evaluation metrics such as accuracy, precision, recall, and f1 score. After illustrating the concept and formula for each evaluation metric, we presented the LSTM results on training, validation, and testing sets which all were above 94%. Finally, the LSTM model is expected to manage to classify future job ads into real and not relate to ads.

# Chapter 7

## 7.1 Introduction

The conclusion and future work chapter is a crucial part of our project report as It provides an overview of the results obtained from the study, highlights the limitations of the study, and provides suggestions for future research.

The conclusion section should summarize the study's key findings and provide an interpretation of the results. It also reflects on the study's limitations too, including the extent to which the results can be generalized to other populations or contexts.

The future work section will outline potential avenues for further research and suggest improvements or extensions to the current study.

## 7.2 Conclusion

In this section, we will summarize our work during the whole project. We tried to analyze a wide range of job ads that were collected or scrapped via Twitter API. Our main goal was to build a deep learning model that learns from historical tweets to be able to classify future ads either real or not.

Our main model was the Long Short-Term Memory (LSTM) which requires some preprocessing steps to be applied to our dataset. These steps include tweet cleaning, splitting the dataset into training and testing, and converting tweets from the textual format into numeric. Our LSTM model managed to reach plus 90% according to all evaluation metrics including accuracy, precision, recall, and f1 score.

Finally, we performed a test using a simple user interface while the user enters a tweet or job ad and the model predicts if it is real or not, furthermore displaying all ads that contain the same tweet.

## 7.3 Difficulties and Limitations

In this section, we will present the limitations that face during our project. As with all deep learning projects, the dataset plays a major role in model performance and results. Our dataset focuses on Arabic ads which was a big challenge as most job ads are written in Arabic. Also, the size of the dataset is important, we managed to collect only 2000 job ads which may be considered a low number according to the recommendations of the deep learning society.

Deep learning models usually are complex and have a huge number of parameters that requires a high computational power so, when we tried to add a complexity to our model and add extra layers which may help improving the model performance but we face an issue regarding computational resources as LSTMs are computationally expensive and require a lot of memory and processing power.

## 7.4 Future Work

This chapter will conclude our recommendations to improve the model performance and reduce the generalization error. As an initial step, in the next research may collect more data and add variety to it by collecting data from many sources such as job platforms like LinkedIn and Wuzzuf.

Secondly, the hyperparameter tuning process. The current model may have suboptimal hyperparameters, and further tuning could improve performance. Grid search or random search techniques can be used to find the optimal hyperparameters.

Further feature engineering can be done to see if additional information can be added to the input data to improve the model's accuracy. For example, including information about the company or the industry can provide additional context to the job ads.

Finally, exploring different architectures: Different architectures of LSTM can be examined to see if they perform better on the job ads classification task. For example, using a bidirectional LSTM or adding attention mechanisms can be evaluated, or using completely different models such as Transformers.