# The University of Jordan

## Authorization Form

I, Mays Suleiman Qudah, authorize the University of Jordan to supply copies of my Thesis/ Dissertation to libraries or establishments or individuals on request, according to the University of Jordan regulations.

Signature:

Date:

# A Hybrid Machine Learning Approach to Extract Sentiment from Arabic Tweets

By

Mays Suleiman Faleh Qudah

Supervisors

Prof. Gheith Abandah

Co-Supervisor

Dr. Fahed Jubair

This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master's Degree of Computer Engineering

Faculty of Graduate Studies

The University of Jordan

May, 2022

# COMMITTEE DECISION

**This Thesis/Dissertation (A Hybrid Machine Learning Approach to Extract Sentiment from Arabic Tweets) was Successfully Defended and Approved on ----------**

| **Examination Committee** | **Signature** |
|---|---|
| Dr. Gheith Abandah, (Supervisor)<br>Professor of Computer Engineering | ---------- |
| Dr. Fahed Jubair, (Co-Supervisor)<br>Assistant Professor of Computer Engineering | ---------- |
| Dr. Ramzi Saifan (Member)<br>Associate Professor of Computer Engineering | ---------- |
| Dr. Mohammad Abdel-Majeed, (Member)<br>Associate Professor of Computer Engineering | ---------- |
| Prof. Ali Al-Haj, (Member)<br>Professor of Computer Engineering<br>(Princess Sumaya University for Technology) | ---------- |

# AKNOWLEDGEMENT

I would like to express my gratitude to my thesis supervisors, Prof. Gheith Abandah and Dr. Fahed Jubair who guided me throughout this thesis and helped me finalize it. I would like also to thank my family who supported me every step of the way.

# Table of Contents

# List of Tables

# List of Figures

# List of Abbreviations and Symbols

| Abbreviation | Clarification |
|---|---|
| BERT | Bidirectional Encoder Representations from Transformers |
| CNN | Convolutional Neural Network |
| GCN | Graph Convolutional Network |
| LSTM | Long Short Term Memory |
| MLM | Masked Language Model |
| MSA | Modern Standard Arabic |
| NER | Named Entity Recognition |
| NLP | Natural Language Processing |
| NSP | Next Sentence Prediction |
| PMI | Pointwise Mutual Information |
| RNN | Recurring Neural Network |
| SVM | Support Victor Machine |

# A HYBRID MACHINE LEARNING APPROACH TO EXTRACT SENTIMENT FROM ARABIC TWEETS

**By**

**Mays Suleiman Faleh Qudah**

**Supervisor**

**Prof. Gheith Abandah**

**Co-Supervisor**

**Dr. Fahed Jubair**

## ABSTRACT

Sentiment analysis for Arabic text is an area with vast interest in the recent years. Several methods have been used in Arabic sentiment analysis. However, the proposed models' ability to capture global information about the data is limited to the context of the sentence information. In this thesis, we illustrate the strength of using graph convolutional network (GCN) in capturing global meaning and we combine the graph with MARBERT, which is one of the most advanced models in Arabic language processing, to achieve higher accuracies for multiple Arabic Twitter datasets. The GCN-MARBERT model utilizes the MARBERT tokenization method. It builds a graph neural network that depends on word pointwise mutual information to build a relation between the corpus vocabulary and this graph. This graph is passed through attention mechanism to the model's classifier to achieve the classifications of the tweets while maintaining the global information about the vocabulary. The proposed

model was trained on three datasets and achieved the highest accuracy on these datasets compared to the state-of-the-art models. The datasets used in this work are ASTD, AJGT, and ArSarcasm datasets. The achieved accuracies for these datasets are: 74.2%, 97.2%, and 72.4%, respectively. This demonstrates the efficiency of using graph neural networks in sentiment analysis and text classification and its potential as a base model for Arabic language processing tasks.

# Chapter 1: Introduction

In this introduction chapter the background of the sentiment analysis in Arabic language is discussed. In addition, we will explain the research problem and significance. Afterwards we will discuss the research hypothesis in addition to the thesis outline.

## 1.1 Background

The emergence of social media has given web users a venue for expressing and sharing their thoughts and opinions on all kinds of topics and events. Twitter, with nearly 200 million active users and over 500 million tweets per day, has quickly become a gold mine for companies and brands to monitor their reputation by extracting and analyzing the sentiment of the Tweets posted by the public about them, their markets, and competitors.

Sentiment analysis or opinion mining is the classification of given text subjectively to certain opinions or sentiments. The sentiment could either be a positive or a negative sentiment.

The most common systems for sentiment analysis address a single language, usually English. However, with the growth of the Internet around the world, users write comments in different languages. To analyze data in different languages, multilingual sentiment analysis techniques have been developed (Dashtipour *et al*. 2016). With this, sentiment analysis frameworks and tools for different languages are being built.

Arabic language is a rich language, and it is widely used in social media in many formats. The modern standard Arabic language is used in formal speech and transactions in media and news. There is also classical Arabic which is used literary and poetic texts in addition to holy texts. And there are also the public dialects where each region in the Arab world has its own dialect.

Natural Language Processing (NLP) or Computational Linguistics is part of the computer science and a branch of artificial intelligence. NLP tools analyze written texts automatically without human intervention. The final aim of NLP is to enable machine to understand human language. There are many techniques for Arabic natural language processing such as normalization, tokenization, named entity recognition (NER), part of speech tagging (POS) and Stemming (Kanan *et al*. 2019). These preprocessing techniques can be performed on Twitter data while also using various word-embedding models like: Word2Vec, FastText, Universal Sentence Encoder to process the source text before passing it to the machine learning model.

There are many approaches used in machine learning to extract sentiment and polarity from source text. Machine Learning techniques for sentiment analysis require the use of a training set and a test set for classification. Training set contains input feature vectors and their corresponding class labels. Using this training set, a classification model is developed which tries to classify the input feature vectors into corresponding class labels. Several machine learning techniques were to developed to classify polarity. The classification methods such as Naive Bayes, Maximum Entropy, and Support Vector Machines (SVM) are used to classify polarity. In recent years, deep neural networks have been largely adapted for text classification such as Convolutional Neural Network (CNN) and Recurring Neural Network (RNN) and especially long short term memory (LSTM).

The emergence of using pre-trained models in machine learning has caused a major improvement on language understanding tasks. The models that first started with BERT model was adapted to Arabic language by training it on Arabic corpus and emerged two

important models for standard Arabic and dialect Arabic. These models are ARBERT and MARBERT (Abdul-Mageed *et al*., 2020).

Despite the huge improvements in understanding Arabic text that is achieved by these models their understanding of context is restricted to the same sentence. The new research for graph convolutional network (GCN) and its use in natural language understanding (Kipf and Welling, 2017) used the relational structure of the graph models to achieve global understanding of the text information.

In this thesis we propose a model for Arabic language that combines these two state of the art methods MARBERT and GCN to optimize the strength of these methods. We built a graph based on the vocabulary and feed this graph to MABERT encoder. The graph and MARBERT interact together so that the classifier will get the global information about the vocabulary from the graph in addition to the local information from MARBERT so that the final representation is used to achieve the sentiment classification of the Arabic tweet.

**1.2 Research Problem**

Most sentiment analysis approaches in Arabic currently are based on word occurrence frequencies. First, the frequency of the words containing polarity is extracted, and then the overall polarity of text is determined. However, determining the overall polarity of the sentence is difficult when using these approaches because we need to consider the word order and the relation between words. For example, in "مع أن الفيلم مكرر الا انه يجذب المشاهد للمتابعة" though the first part of the sentence expresses a strong negative sentiment, but the overall polarity is positive. So we need a method that takes into consideration the order in addition to the context of the meaning that is expresses in an implicit way.

### 1.3 Research Significance

To address the limitations of determining the polarity in complex Arabic sentences, we propose a framework that integrates word embedding from MARBERT and a vocabulary graph built on word co-occurrence to analyze the source text. This is to improve the overall performance and accuracy of polarity detection for complex sentences.

### 1.4 Research Hypothesis

The research supposes that the sentiment of Arabic tweets can be classified to multiple sentiment classification, such as positive and negative polarity and that the polarity of the tweets can be determine by building a graph of Arabic vocabulary and create relation between words and pass this graph network integrated with transformer model.

### 1.5 Thesis Outline

The thesis consists of six chapters. The second chapter discusses related work to sentiment analysis and related to graph convolution in addition to its relation with BERT based models. In Chapter three, the theoretical aspect of the related models is discussed where it includes detailed description of transformer-based models and graph convolutional networks. Chapter four presents the experimental setup, gives a description of the used datasets, and describes the research methodology. In Chapter five, I include my results and compare them to state-of-the-art solutions. Finally, in Chapter six, I discuss the research conclusions and future work.

# Chapter 2: Related Work

There are numerous studies that have discussed Arabic sentiment analysis. In the following paragraphs, we review some of these studies. We will talk about the traditional methods of sentiment analysis and state of the art models for Arabic language then we will talk about the previous work that used GCN in text classification.

## 2.1 Arabic Models for Sentiment Analysis

Sentiment analysis for Arabic language is an area of interest for researchers and its interest has increased in recent years. We will start with Shoukry (2012) who used an approach of feature vectors and applied it to the Naive Bayes and Support Vector Machine (SVM) Classifiers with the aim of comparing the results and choosing the classifier with the higher accuracy and concluded that SVM has the highest accuracy in determining the polarity of Twitter data with accuracy of 72.6%. Neethu *et al.* (2013) proposed a method to analyze twitter posts about electronic products like mobiles, laptops etc. using Machine Learning approach. They suggested that by doing sentiment analysis in a specific domain, it is possible to identify the effect of domain information in sentiment classification. A new feature vector was presented for classifying the tweets as positive, negative and extract peoples' opinion about products. The highest accuracy reached 90%. Duwairi *et al.* (2014) suggested applying three classifiers on an in-house developed dataset of tweets/comments. In particular, the Naïve Bayes, SVM and K-Nearest Neighbor classifiers were run on this dataset. Their results show that SVM gives the highest precision with 75.25% while KNN (K=10) gives the highest Recall of 69.04%. Gautam *et al.* (2014) used a method to pre-process the data, after that extracted the adjective from the dataset that have some meaning which is called feature vector, then selected the feature vector list and thereafter applied machine learning based

classification algorithms. The algorithms used in this study are mainly Naïve Bayes, Maximum entropy and SVM. The Semantic Orientation based WordNet was also used which extracts synonyms and similarity for the content feature. The highest accuracy reached by WordNet reached 89.9%. Aldayel *et al.* (2015) suggested a hybrid approach that combines semantic orientation and machine learning techniques. Through this approach, the lexical-based classifier will label the training data, a time-consuming task often prepared manually. The output of the lexical classifier will be used as training data for the SVM machine learning classifier. The experiments show that hybrid approach improved the accuracy by 16.41%, achieving an overall 84.01%.

Altowayan and Tao (2016) proposed a word embedding method for Arabic tweets and used multiple classifiers on the word embedding to train the data. The resulting of which is that using word embedding achieved higher results. The highest accuracy was 80% on ASTD-ArTwitter dataset using logistic regression classifier.

Alomari *et al.* (2017) investigated different supervised machine learning sentiment analysis approaches when applied to Arabic user's social media of general subjects that are found in either Modern Standard Arabic (MSA) or Jordanian dialect. Experiments are conducted to evaluate the use of different weight schemes, stemming and N-grams terms techniques and scenarios. The results showed that SVM classifier using term frequency–inverse document frequency (TF-IDF) weighting scheme with stemming through Bigrams feature outperforms the Naïve Bayesian classifier. The accuracy for deep neural network model reached 85%. And reached 90% for convolutional neural network model. There are many models that used a hybrid approach; for example, HILASTA (Elshakankery and Ahmad, 2019) used a hybrid model that combines lexicon based and machine learning

models. The model depends on auto learning and updating the lexicon and it achieved the highest results using SVM classifier.

Some papers used Convolutional neural network (CNN) for sentiment analysis. Heikal *et al*. (2018) proposed a CNN model and combined it with LSTM to predict the sentiment of the tweet. Dahou *et al* (2019) suggested combining differential evolution algorithm with the CNN model. In this paper the differential evolution algorithm is automatically used to search for the configuration that is most optimal which includes CNN architecture and parameters. This achieved higher results than CNN based models. Mazajak model (Farha and Magdy, 2019) is a web tool used for sentiment analysis. This tool also used CNN and LSTM in its model. Ombabi (2020) Proposed a deep learning model for Arabic language sentiment analysis based on one-layer CNN architecture for local feature extraction, and two layers LSTM to maintain long-term dependencies. The feature maps learned by CNN and LSTM are passed to SVM classifier to generate the final classification. This model is supported by FastText words embedding model.

### 2.2 ARBERT and MARBERT

The emergence of transformers that are pre-trained started with BERT Devlin *et al*., 2019) is a turning point in sentiment analysis and language processing tasks. The BERT model achieved state of the art results on all NLP tasks and its strength is it can be fine-tuned to any dataset to achieve better results.

Antoun *et al*. (2020) developed AraBERT which is a pre-trained transformer on Arabic data it achieved state of the art results on Arabic NLP tasks. Abdul-Mageed *et al.* (2020) introduced two models ARBERT and MARBERT that have superior performance to all existing models. When fine-tuned on Arabic data, ARBERT and MARBERT collectively

achieved highest results (Abdul-Mageed *et al.,* 2020). ARBERT is trained on standard Arabic and MARBERT is trained on dialect. These models were adapted and fine-tuned in other researches. In Camel tools (Obeid *et al*. 2020) MARBERT has been fine tuned to provide a python toolkit to support NLP tasks on Arabic language using APIs and command line interface. DeepBlueAI (Song *et al*., 2021) used MARBERT as an encoder to output the sentence embeddings. After that the embeddings are input into a separate attention layer and an output layers. Abdel-Salam (2021) proposed two models one that used fine-tuned MARBERT and the other is built by LSTM and CNN. The best performing model in this paper is the MARBERT model. Wadhawan (2021) fine-tuned AraBERT model in order to use the model for sentiment analysis.

QARiB model (Abdelali *et al*., 2021) is a proposed method of pre-trained five different models of BERT and was trained on a mixed data of standard and dialectical Arabic.

### 2.3 GCN in Text Classification

Graph convolutional network is recently receiving growing attention and there has been new research to show its use in text classification in general and in sentiment analysis in specific. BertGCN (Yao, 2019) is trained by using GCN with BERT based model this optimizes the use of large scale model and graph based method. TextGCN (Lin *et al*., 2021) also uses GCN with a graph built from text and connect the words as nodes where the edges are calculated by the weight of the word co-occurrence algorithm. VGCN-BERT (Lu *et al*. 2020) augmented BERT model with graph embedding to achieve state of the art results for text classification.

# Chapter 3: Neural Networks for Sentiment Analysis

In this chapter the various machine learning model used in sentiment analysis is discussed. Where first we discuss BERT model and its Arabic language based model: ARABERT and MARBERT. Then Graph Convolutional Networks are discussed and how they are used as model for sentiment analysis.

## 3.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) is one of the major recent achievements in the machine learning world. It achieved state of the art results for a variety of NLP (Natural Language Processing) tasks. This includes tasks such as Question answering, Text classification and others.

The main innovation key in BERT is by applying the training in a bidirectional way; whereas previous methods usually worked in a unidirectional way i.e. they used to read the words sequentially. This new method was enabled by introducing the bidirectional transformers. Transformers are a deep learning method that works by connecting each element by each other and their weighting is then calculated dynamically by the connections which is what's called attention.

Applying the multi-layer bidirectional method on BERT achieved a deeper understanding of the sentences meaning and words context. This method works by applying two main tasks: Masked language model and Next sentence prediction.

### Masked Language Model (MLM)

Masked Language Model (MLM) is the task where a sentence is given to the BERT model, and the model is expected to predict the next word. This is done by masking the

sentence with a MASK token where usually 15% of the words are masked. After that the model works on predicting what the masked word is. This prediction is done by learning linguistic patterns and predict the word from the sequence context.

**Next Sentence Prediction (NSP)**

The second task in BERT model is the Next Sentence Prediction (NSP). This task aims to make long term understanding between sentence where MLM is used to understand relation between words.

In this task BERT is given two sentences: A and B and is asked to classify if sentence B is a subsequent for sentence A. since half the input is consisted of pair of sentences and the other half is random sentences from the whole corpus. In order for the model to distinguish the different sentences from each other each sentence will be tokenized and two special tokens are added to the sentence. The token [CLS] is added to the start of each sentence and the token [SEP] is added between each two sentences.

After that two embedding methods are added to each token: sentence embedding and positional embedding; where sentence embeddings indicates token belongs to which sentence and positional embedding represents the position of the token in the sequence. Figure 1 shows the different types of embedding in a BERT model (Devlin *et al*, 2018).

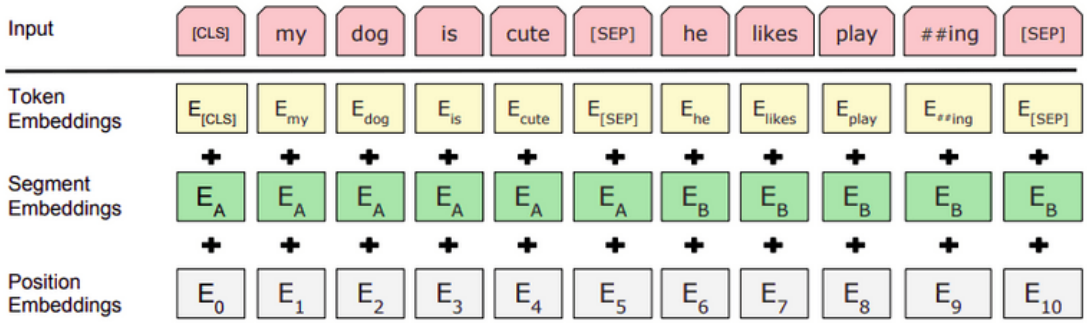


Figure 1 Bert Input Embeddings (Devlin *et al*., 2018)

When the sequence is passed to the transformer model, the probability of the value of IsNext label will be calculated and will be calculated with softmax where the output is predicted accordingly. Figure 2 shows the structure of BERT pre-training process (Devlin *et al*, 2018) that will use the two unsupervised tasks: MLM and NSP.

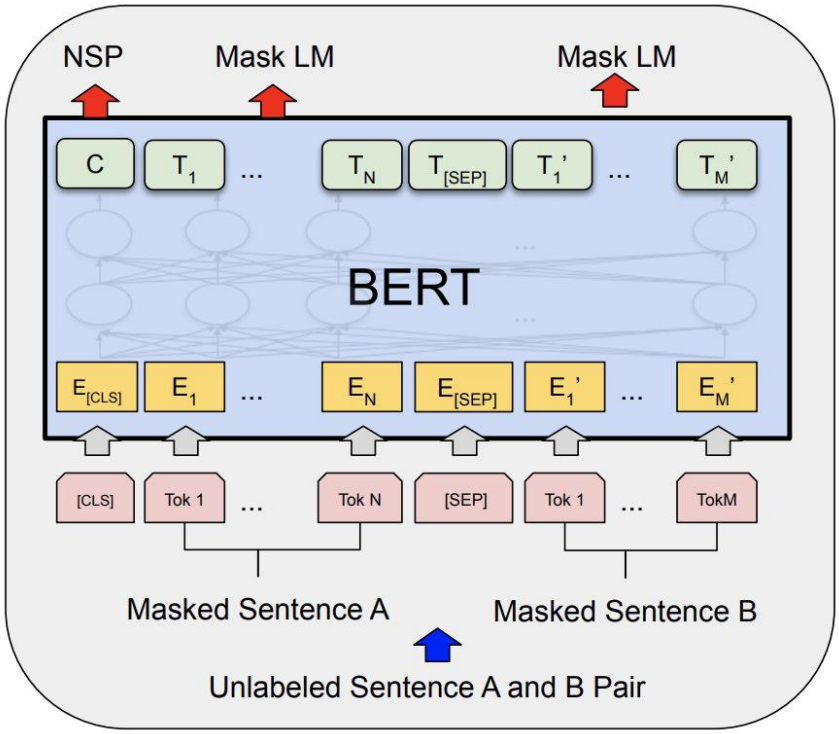


Figure 2 BERT Pre-training (Devlin *et al*., 2018)

This is how the general BERT model is built. In order to use this model for a wide variety of NLP tasks by modifying this model and adding a layer to it. This process is called fine-tuning. In this process tuning the hyper parameters will result in achieving state of the art results for various challenging tasks.

In Figure 3, fine tuning BERT model for sentence classification (Devlin *et al*, 2018).

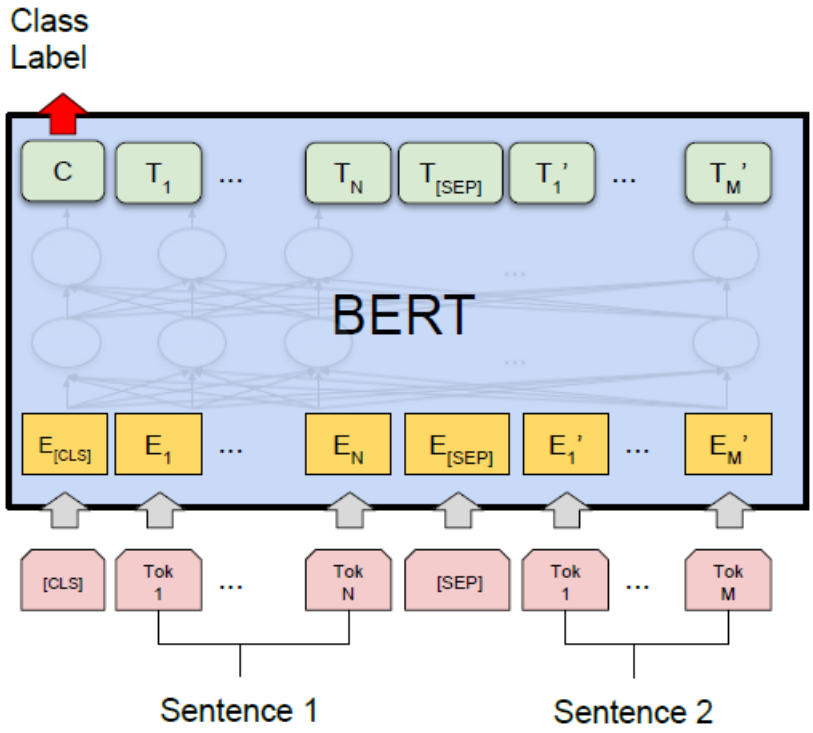


Figure 3 BERT Fine-tuning for Sentence Classification

### 3.2 ARBERT and MARBERT

Multilingual models have been built based on BERT model where large amount of data is used for the pre-training process. In this section two of the most important models for Arabic language are discussed.

### 3.2.1 AraBERT

Arabic Language BERT based model which is named AraBERT is a model for large scale and a pre-trained masked language model that focuses on Modern Standard Arabic (MSA). ARBERT is trained using the same architecture as BERT which are twelve attention layers each layer with seven hundred and sixty-eight hidden dimension. The vocabulary used for training is 70 million sentences that covers Arabic news websites and two public domain MSA datasets (Antoun *et al*., 2020).

### 3.2.2 MARBERT

Most of the Arabic content and especially on social media platforms is not written in MSA. It is rather written in many dialects so using ARBERT is not suitable to treat these data. MARBERT is the BERT based Arabic model that is trained on large Twitter datasets to achieve state of the art results for dialect based Arabic NLP tasks (Abdul-Mageed *et al*, 2020). This model is trained using the same architecture as ARBERT. The dataset used for training is a huge Twitter dataset consisting of 128 GB of text size which is more than fifteen billion tokens.

## 3.3 Graph Convolutional Networks (GCN)

Graphs are widely used in various real type applications as a way of structuring data. We can see graph representation in many applications such as prediction and social networking. The structural representation of graph enables building a relational model that combines the information known about data. Deep learning techniques are applied to graphs in a way that is different than the usual deep learning method that works on Euclidean space. One of the most important algorithms of graph neural networks is Graph Convolutional

Networks (GCN). In general, a graph is a structure that consists of nodes and edges. This is denoted by:

$$G = (V, E)$$

Where $V$ stands for vertices or nodes and $E$ represents edges between the nodes. In Figure 4, a simple graph is represented with its nodes and edges.

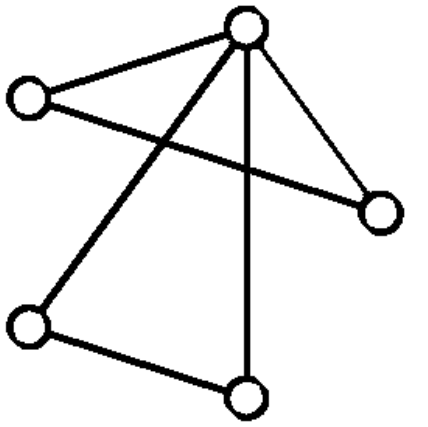

Figure 4 Simple Graph Representation

Graph is also called a weighted graph if there is a weight calculated for each edge. In GCN the words are converted to node features that are all connected in a graph. Each node gets the information from all its node neighbors and the result of this information is fed to a neural network. In GCN each node knows information about its neighbor using one layer of convolution, in order to achieve more knowledge about its surroundings multiple layers of convolution are applied to the graph. Figure 5 illustrates how stacking multi layers of graph convolution includes more global information. In multi-layer convolutional neural network the output of the first layer is the input of the next layer. We can define the number of layers as how far of a distance the node features can transport. At first each node

within a one-layer graph network can only get information about its neighbors. When another layer is stacked on top of the first layer, the collecting information process is repeated per each layer. However, in this stage each stacked layer already has the information about its neighbors. So we configure the number of layers of how deep we want the node to travel to get information about the network.

Most GCN models use three to four layers.

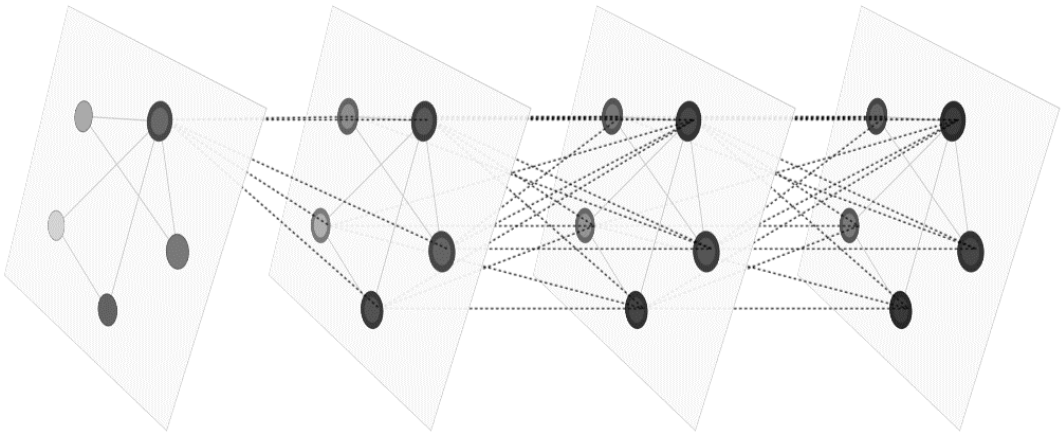

Figure 5 Multi-Layer Graph Neural Network

We can stack more layers to make GCNs deeper. Consider residual connections for deep GCNs. Normally, we go for 2 or 3-layer GCN.

# Chapter 4: Experimental Setup and Methodology

In this chapter the environment setup is detailed, afterwards we talk about the used datasets. Preprocessing method is detailed as well as the model details; where we discuss building the graph model and integrating it with MARBERT. Lastly we talk about the experiment method and the research methodology.

## 4.1 GCN-Bert Hybrid Model

In this section, the training model is explained in detail. From building the word graph and creating relationship between words, to passing the graph to embedding layers that will be passed to the pre-trained BERT model.

The following **Error! Reference source not found.**5 represents the flow of the model and the next sections will contain more details about the model.

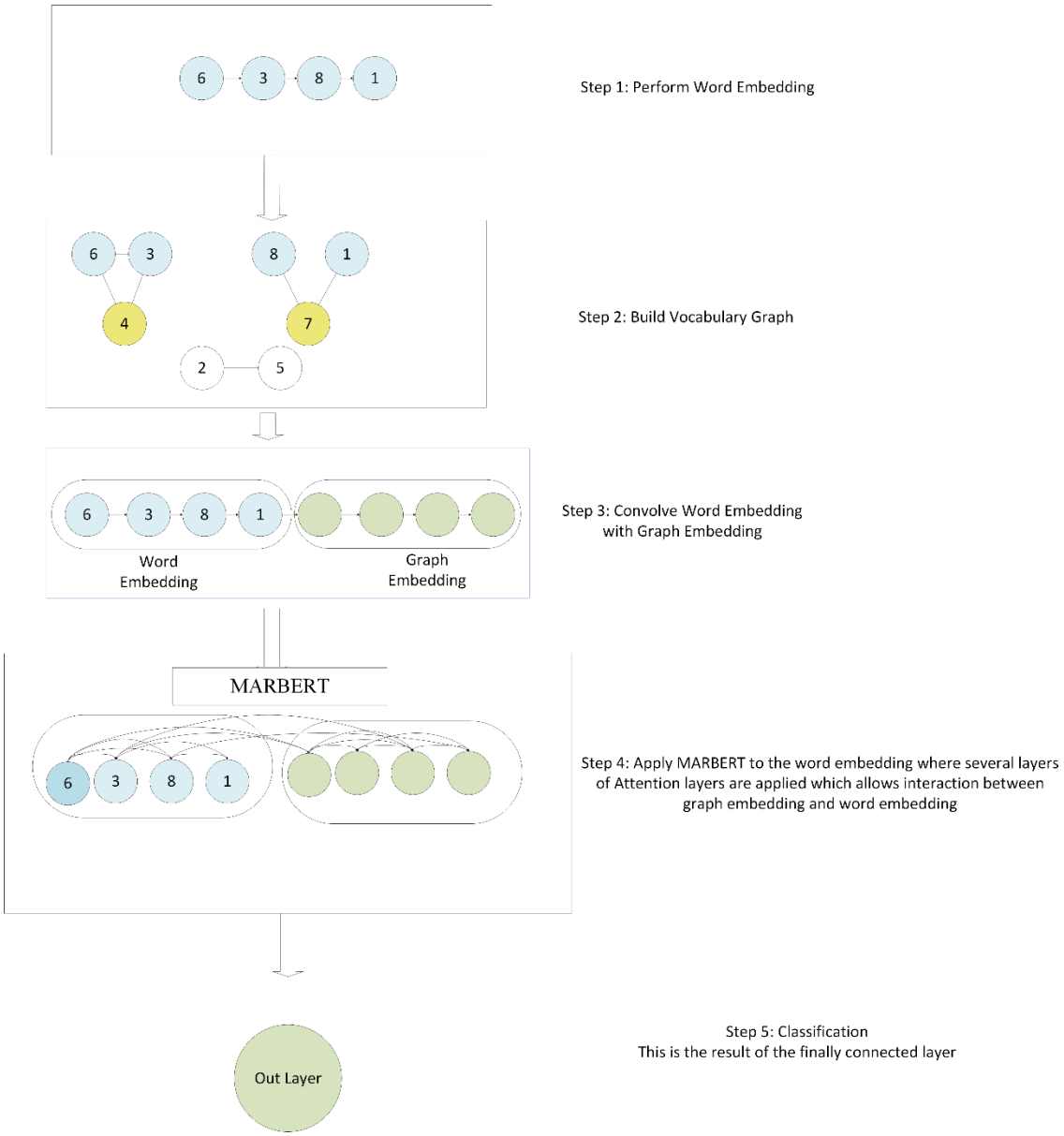


Figure 6 GCN BERT Model Overview

### 4.1.1 Building the Vocabulary Graph

After preprocessing the data as discussed in section 4.3, the graph model is built by computing the word embedding using the pointwise mutual information (PMI) algorithm as described in Equation 1. The relation between each pair of words is calculated between a range of numbers that the PMI is normalized to. So the range of values for word correlation is set between -1 and 1. When the values approaches positive 1 then this means high correlation between the pair of words and if the value approaches -1 then there is no or little correlation between the words. To make sure that the dependency between words is kept, the calculations between words that are in the same sentence is kept in a value that represents the sliding window size and the size of the sliding window is the size of the sentence.

Now the graph is constructed where each node represents a word or a vocabulary from the overall words in the corpus and the edge is created when there is a correlation between words. We set a threshold value for the edge, so that the edge is created when the pmi value exceeds this edge.

$$\text{PMI} = \frac{log\ p(i,j)}{p(i)p(j)} \qquad (1)$$

### 4.1.2 Graph Convolutional Model

The general method to use a graph convolutional network (GCN) is a universally similar, where the network is a multilayer network usually two layers and their vectors are induced on nodes in relation to their neighbors. So the general graph is represented by:

$$G = (P, E) \qquad (2)$$

And the general function for GCN is calculated by:

$$H = AXW \qquad (3)$$

Where $A$ is the adjacency matrix, $X$ is the input matrix and $W$ is the weight matrix.

In our model we define $A$ as the vocabulary graph matrix and $X$ as the input matrix that contains the vocabulary related to the input sentence. And W is the weight for the hidden state for the document. So when we apply equation 1 to our model where there are two layers for GCN and the activation layer is ReLu we will get the following equation

$$H = ReLU(XAWvh)Whc \qquad (4)$$

Where $v$ is for vocabulary size and $h$ represents the hidden layer size and $c$ is the class size or embedding size. Which means $Wvh$ is the weight of the input sentence in the hidden layer combined with $Whc$ is the weight of the class size or the whole sentence. This aims to combine the weight of input related to the words in the sentence and convolves it in the next layer with the weight of the whole graph. In this method we achieve capturing the meaning related to the whole document instead of the relatively local meaning of the word.

4.1.3 Integrate Graph Convolutional Model with MARBERT

In this step we convolve the graph embedding that is obtained from the previous section and sequence of word embedding from MARBERT pre-trained model, and calculate the attention score.

The attention score is passed to the hidden layers of the model which are 12 hidden layers and will keep the information of the relation between words for local embedding which is achieved from GCN embedding and global embedding which is achieved from sequence

embedding and all layers are then activated by ReLu layer that will obtain the final embedding output.

An example of how the GCN-Marbert model achieves deeper understanding of Arabic text is the sentence: "مع ان الفيلم مكرر الا انه يدفع المشاهد للمتابعه حتى النهايه". The previous sentence contains an implicit positive sentiment. So, the model trains the previous sentence and by using the attention layers concatenated with the GCN embedding it will be able to understand that the latter part of the sentence is actually related to other words that have positive meaning which will be classified as a positive sentence at the final step of classification.

### 4.2 Experiment Method

The experiment is executed on three datasets mentioned in Section 4.5. The experiment aims to predict the sentiment for Arabic Tweets. The state of the art methods use Bert based methods especially ARBERT and MARBERT models. MARBERT achieves the highest accuracy where it comes to twitter datasets.

In this experiment, the model is a hybrid model that combines MARBERT with Graph Convolutional Network with the aim to increase sentiment detection accuracy.

The hyper-parameters used in this experiment are as follows:

Batch size, learning rate, max sequence length, dropout rate and loss weight decay.

The model is trained with 10 epochs.

### 4.3 Methodology

The following steps describe the methodology used in this thesis starting from the survey process and ending with the thesis documentation.

1. Survey of related work: review the related work that deals with Arabic sentiment analysis.
2. Evaluation of related work: show the results of the related work and outline the tools that been used.
3. Dataset gathering and preparation: collect the datasets using Twitter API and classify the tweets manually.
4. Clean the dataset and apply word preprocessing.
5. Prepare the Vocabulary Graph.
6. Combine Phase 2 and 3 to produce Graph Enriched Embedding
7. Concatenate with Marbert
8. Fine tuning and accuracy improvement: test the model and fine tuning it to achieve better accuracy.
9. Documentation: document our work and show the results.

### 4.4 Experimental Setup

The following table shows the environment used in the experiments.

The table includes CPU type, memory, OS, and programming language used and associated libraries.

Table 1 Environment Specifications

| Components | Specifications |
|---|---|
| CPU | Intel Core i7-1165G7 CPU @ 2.80GHz<br>CPUs (Cores): 8<br>L1dcache: 320KB<br>L2 cache: 5MB<br>L3 cache: 12MB |
| Memory | 24 GB DDR4-SDRAM @ 3200MHz |
| Libraries | Python 3.8.12<br>PyTorch: 1.5.0<br>scikit-learn: 0.24.2<br>NumPy: 1.19.2<br>SciPy 1.4.1 |
| OS | Windows 10 Enterprise, 64-bit |

### 4.5 Datasets

This section describes the various datasets used in this research and how these datasets are prepared for the training of the model.

#### 4.5.1 ASTD Dataset

Arabic Sentiment Tweets Dataset (ASTD) is Arabic language datasets that contain 10000 Arabic tweets. The dataset has four classes:

Positive, Negative, Objective and Neutral.

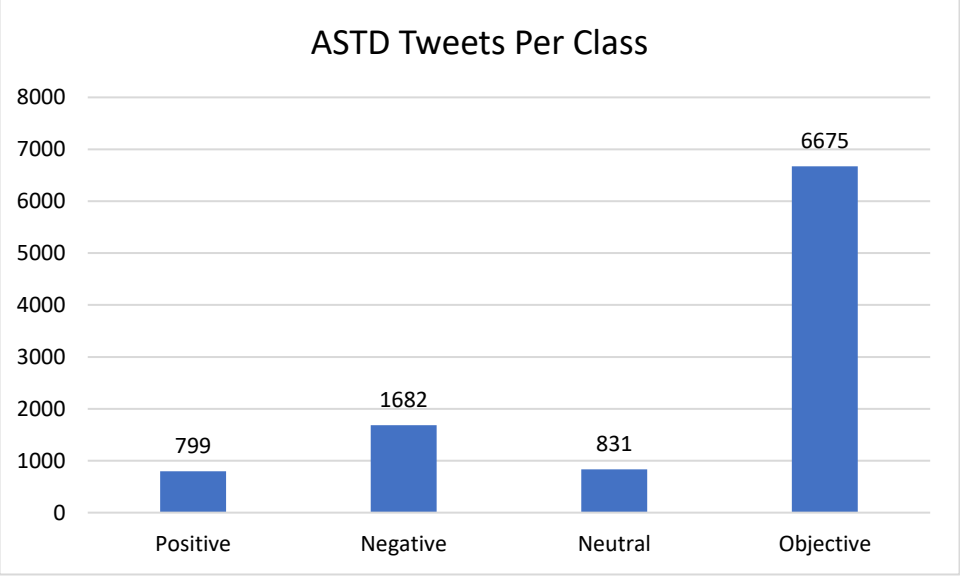


Figure 7 ASTD Dataset Classes

To prepare the dataset, the objective tweets are eliminated from training process and three classes are used. Also, for the experiment the dataset was split into 3 sets; train, validation and test sets with the ratio 60:20:20.

### 4.5.2 AJGT Dataset

Arabic Jordanian General Tweets (AJGT) is an Arabic language dataset that contains tweets of Jordanian dialect. The dataset contains 1800 tweets.

The dataset has two classes: Positive and Negative.

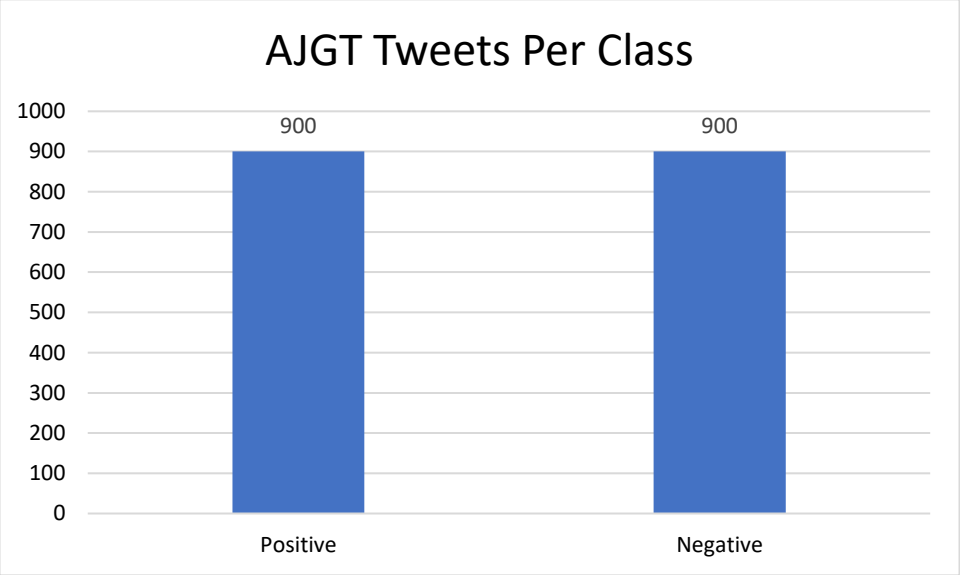


Figure 8 AJGT Dataset Classes

In this experiment the dataset set was split to train set and test set at the ratio of 80:20.

### 4.5.3 ArSarcasm V.2 Dataset

ArSarcasm-v2 dataset is a new version from the ArSarcasm dataset along with portions of A Dialectal Arabic Irony Corpus Extracted from Twitter (DAICT) corpus. Each tweet is collected with its sarcasm, sentiment and dialect. The overall tweets in the dataset are 15,548 tweets. ArSarcasm-v2 is used in many papers but was first released as the official dataset in WANLP 2021 Shared Task on Sarcasm and Sentiment Detection.

The dataset contains the following fields:

Tweet, sarcasm, sentiment and dialect. In this experiment the following fields are used for sentiment detection:

- Tweet: which represents the text of the tweet and which will be used to build the graph model.
- Sentiment: which represents the sentiment of the tweet and has three classes (positive, neutral and negative)

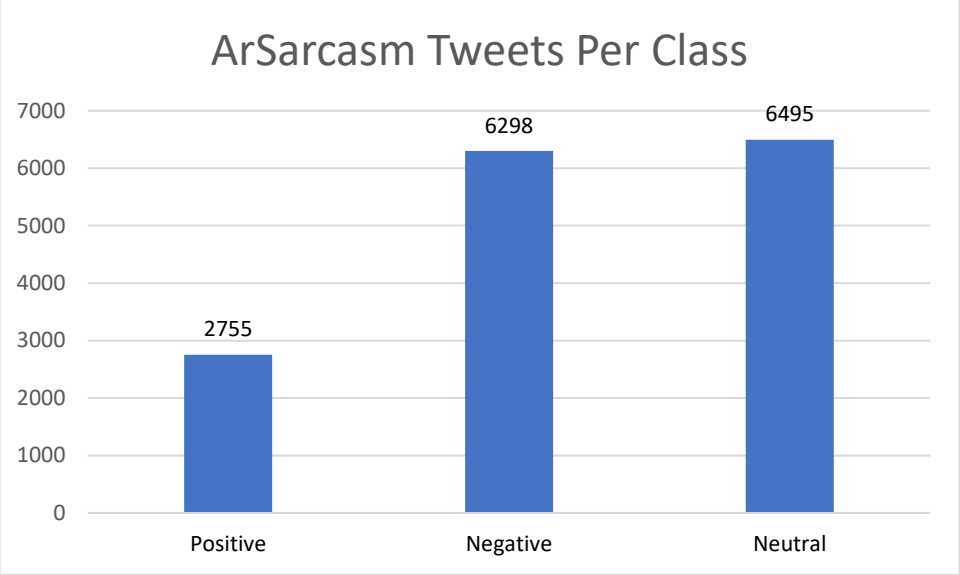


Figure 9 ArSarcasm Dataset Classes

### 4.6 Preprocessing

In the preprocessing stage we prepare the data by applying a method to clean the text in the datasets and then to tokenize the text.

#### 4.6.1 Normalization

The clean text method used to normalize the Arabic text using AraVec method where the letters “hamza” like (آ ,إ ,أ) are replaced with (ا) and the letter (ة) is replaced with (ه) and the letter (ى) is replaced with (ي).

This steps also removes URLs, mentions, emoticons and emojis. In addition, elongation in words is removed so that when a character is repeated in a word, the extra characters are removed from the word. For example, the word (سلاااااام) is then normalized to (سلام). The reason that we perform this step is to reduce the randomness in the text. This will bring the source text into some sort of a standard data. So that it will help to reduce the amount of different words and information that the machine has to learn, and therefore this will lead to improved efficiency.

### 4.6.2 Tokenization

In this steps the corpus data which contains the dataset (train set, validation set and test set) is tokenized using Bert Tokenizer. The Bert tokenizer used is from MARBERT version of pre-trained Bert.

The sentence is split into words and we make sure by using Bert tokenizer that in the training phase all words are a subset from the pre-trained Bert vocabulary. An example of MARBERT sentence tokenization is illustrated below:

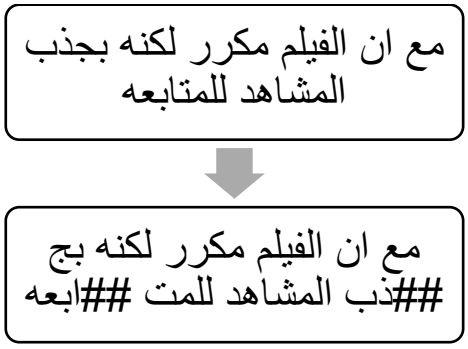


Figure 10 MARBERT Tokenization

# Chapter 5: Results and Discussion

This section discusses the results of the hybrid machine learning approach used in this research and how it compares to previous approaches. We discuss the evaluation metrics used in this experiment and how the model achieved the best results using certain hyper parameters.

## 5.1 Evaluation Metrics

In this section the evaluation metrics that are used to measure the results of the experiment are detailed to see how the model performs in sentiment analysis and classification of Arabic Tweets datasets.

### 5.1.1 Accuracy

Accuracy is the metric used for evaluating classification models. it can be said that accuracy is the percentage of predictions that the model got right. In general, accuracy can be defined as the number of correct predictions divided by the total number of predictions.

### 5.1.2 F1 Score

F1 score is the metric used to calculate the performance of a classifier. It is calculated by calculating the mean between precision and recall values. There are different types of F1 scores: Macro F1, Micro F1 and weighted average F1.

The equation used to calculate F1 score is as follows:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (5)$$

Where Precision is:

$$Precision = \frac{True\ Positive}{True\ Positivie+False\ Positive} \quad (6)$$

And Recall is:

$$Recall = \frac{True\ Positive}{True\ Positivie+False\ Negative} \quad (7)$$

So alternatively, we can write F1 equation as:

$$F1 = \frac{True\ Positive}{True\ Positive+0.5(False\ Negative+False\ Positive)} \quad (8)$$

Macro Average F1 score is calculated by computing the mean score for all classes regardless of their weight. As for Micro F1 it is computed by calculated the sum of all false negative, true positive and false positive values and then pass them to the F1 equation. As for the weighted F1 score it is calculated by multiplying the average F1 score per each class by the value of support of each class; i.e. the number of occurrences per each class. This method of calculating F1 is best suited for unbalanced datasets which is why we used this method in calculating F1 in our research.

### 5.1.3 Loss Function

This metric is used to measure the error in models, the perfect model aims to reach zero loss. In this experiment, Cross entropy loss is used.

## 5.2 Fine-tuning and Hyper Parameter Values

In this Section the hyper-parameters used in this experiment are explained and shows where the best results were received.

### 5.2.1 PMI Threshold

The threshold value for the PMI where graph edge is created between words when this value is achieved. We found that the best result is achieved when the threshold is 0.3.

The below figure shows the accuracy for the ASTD dataset at different threshold values.

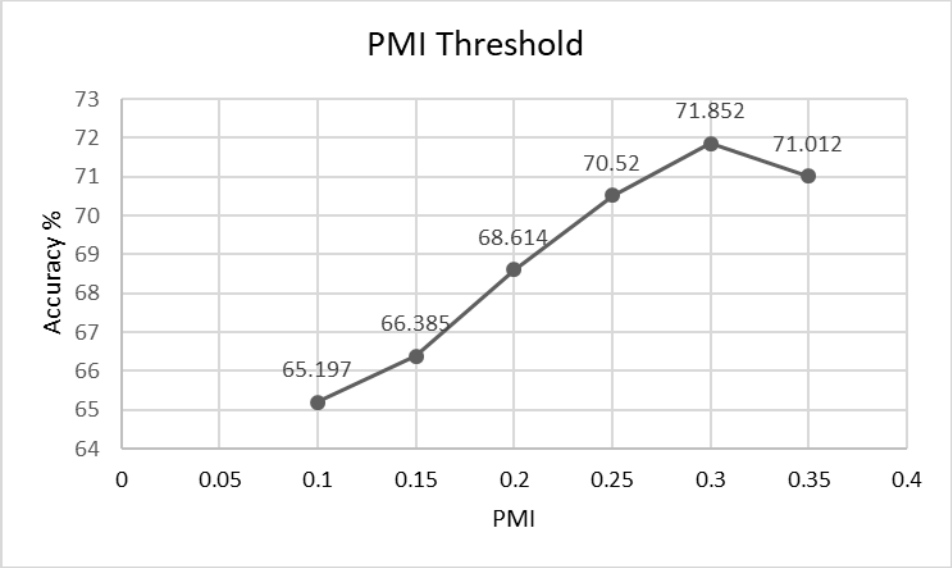


Figure 11 Relation Between PMI Threshold and Model Accuracy

### 5.2.2 Learning Rate

The learning rate is the parameter used to determine the step size used in each iteration in order to optimize the algorithm to achieve the least loss.

In this experiment the learning rate hyper-parameter is of the most importance where the accuracy improved when decreasing this value.

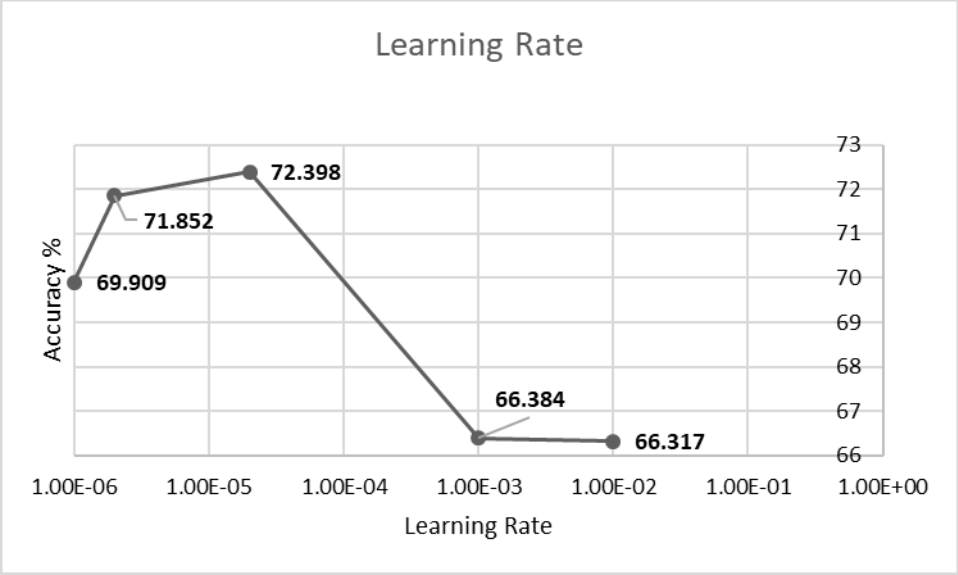


Figure 12 Relation Between Learning Rate and Model Accuracy

### 5.2.3 Batch Size

When changing the value of the batch size used in training the model we noticed that it will have minimal effect on the accuracy. We found that the best value was achieved when the batch size is 32. The following chart shows the effect of changing the batch size on the ASTD dataset.

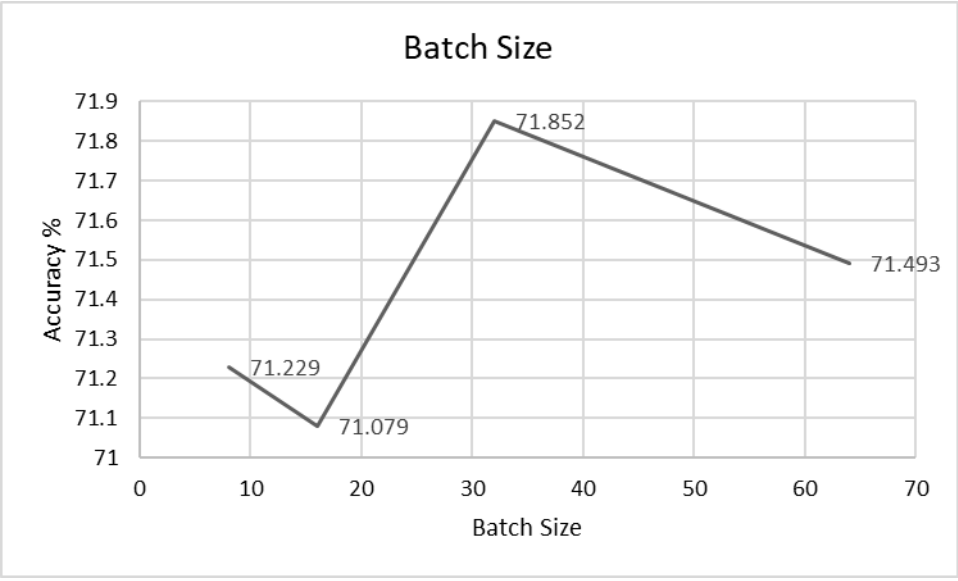


Figure 13. Relation Between Batch Size and Model Accuracy with

### 5.2.4 Optimizer

Optimizers are the algorithm used to achieve the least loss value and are dependable on hyper parameter values such as weights and biases. In this experiment BertAdam optimizer is used as an optimizer function.

## 5.3 Results and Comparisons

The main results on weighted average F1-Score on test sets are presented in the following tables. The main observation is that VGCN-BERT outperforms models that are using the BERT based models alone which highlights the advantage to combine Bert base model with GCN model. It is also observed that combining MARBERT with GCN achieved the best results amongst state of the art models on used datasets.

### 5.3.1 ASTD Results

The results for the ASTD dataset achieved an accuracy of 74.2 and F1 score of 74.7. this is the highest recorded score for this dataset as we compare it with state of the art models. These results are displayed in Table 2.

Table 2 ASTD Comparison Results

| Model | Accuracy | F1 | Notes |
|---|---|---|---|
| ARBERT (Abdul-Mageed *et al*., 2020) | 68.73 | 76.50 | |
| MARBERT (Abdul-Mageed *et al.,* 2020) | 71.02 | **78.00** | |
| hULMonA (ElJundi *et al.,* 2019) | 69.9 | 67.7 | |
| HILATSA (Elshakankery and Ahmad, 2019) | 75.19 | 75.70 | 2 class balanced dataset |
| Mazajak (Farha and Magdy, 2019) | 66 | 72 | |
| CAMeL (Obeid *et al.,* 2020) | N/A | 73 | |
| CNN deep learning (Heikal *et al.,* 2018) | 64.30 | 64.09 | |
| Word Embeddings and logistic regression (Altowayan and Tao, 2016) | 76.66 | N/A | 2 class dataset |
| CNN and Differential Evolution Algorithm(Dahou *et al*., 2019) | 82.48 | 82.57 | 2 class balanced dataset |
| This work (Marbert-GCN) | **74.21** | 74.75 | |

### 5.3.2 AJGT Results

The results for the ASTD dataset achieved an accuracy and F1 score of 97.2 this is the highest recorded score for this dataset as we compare it with state of the art models. These results are displayed in Table 3.

Table 3 AJGT Comparison Results

| **Model** | **Accuracy** | **F1** |
|---|---|---|
| ARBERT (Abdul-Mageed *et al.,* 2020) | 94.44 | 94.43 |
| MARBERT(Abdul-Mageed *et al.* ,2020) | 96.11 | 96.10 |
| QaRiB (Abdelali *et al.,* 2021) | 93.3 | N/A |
| SVM classifier (Alomari *et al*., 2017) | 88.72 | 88.27 |
| CNN and Differential Evolution Algorithm (Dahou *et al.,* 2019) | 93.17 | 93.19 |
| This work (Marbert-GCN) | **97.22** | **97.22** |

### 5.3.3 ArSarcasm Results

The results for the ArSarcasm dataset achieved an accuracy of 72.4 and F1 score of 72.52 this is the highest recorded score for this dataset accuracy and is a very close second to the highest F1 score when compared with state of the art models. These results are displayed in Table 4.

Table 4 ArSarcasm Comparison Results

| **Model** | **Accuracy** | **F1** |
|---|---|---|
| MARBERT with attention interaction layer (Mahdaouy *et al.,*2021 ) | 71.07 | **74.80** |
| DeepBlueAI (Song *et al.,* 2021) | 70.37 | 73.92 |
| Fine-tuned Marbert (Abdel-Salam, 2021) | 69.57 | 73.21 |
| Fine-tuned Arabert (Wadhawan, 2021) | 69.83 | 72.55 |
| This work (Marbert-GCN) | **72.44** | 72.52 |

# Chapter 6: Conclusions and Future Works

In this chapter the conclusion for my results of building a hybrid model to classify Arabic sentiment is discussed as well as the area that future work will focus on.

## 6.1 Conclusions

The use of graph convolutional network for Arabic language is proven to achieve higher accuracies and results. In addition, the model used in this work is a hybrid model that combines MARBERT with GCN proves that the accuracy is better than using BERT based model only.

In conclusion this work achieved higher accuracies than state of the art models; for ASTD dataset I achieved an improvement 3% and in AJGT, I achieved an improvement of 1%. As well as, ArSarcasm dataset achieved 1.3% of higher accuracy.

## 6.2 Future Work

Future work for my thesis is to apply this hybrid model for various NLP tasks other than Sentiment Analysis; e.g. Question answering, and sarcasm detection. The model can also be used to classify other Arabic text in MSA as well as dialectical Arabic language.

In addition, fine tuning and changing hyper parameters proved to achieve higher results for this work. So additional improvements can be achieved by changing their values.

# REFERENCES

Abdelali, A., Hassan, S., Mubarak, H., Darwish, K., & Samih, Y. (2021). Pre-training bert on arabic tweets: Practical considerations. *arXiv preprint arXiv:2102.10684*.

Abdel-Salam, R. (2021, April). Wanlp 2021 shared-task: Towards irony and sentiment detection in arabic tweets using multi-headed-lstm-cnn-gru and marbert. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop* (pp. 306-311).

Abdul-Mageed, M., Elmadany, A., & Nagoudi, E. M. B. (2020). ARBERT & MARBERT: deep bidirectional transformers for Arabic. *arXiv preprint arXiv:2101.01785*.

Aldayel, H. K., & Azmi, A. M. (2016). Arabic tweets sentiment analysis–a hybrid scheme. Journal of Information Science, 42(6), 782-797.

Alomari, K. M., ElSherif, H. M., & Shaalan, K. (2017, June). Arabic tweets sentimental analysis using machine learning. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 602-610). Springer, Cham.

Altowayan, A. A., & Tao, L. (2016, December). Word embeddings for Arabic sentiment analysis. In 2016 IEEE International Conference on Big Data (Big Data) (pp. 3820-3825). IEEE.

Antoun, W., Baly, F., & Hajj, H. (2020). Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Dahou, A., Elaziz, M. A., Zhou, J., & Xiong, S. (2019). Arabic sentiment classification using convolutional neural network and differential evolution algorithm. *Computational intelligence and neuroscience*, *2019*.

Dashtipour K, Poria S, Hussain A, Cambria E, Hawalah AY, Gelbukh A, Zhou Q. Multilingual Sentiment Analysis: State of the Art and Independent Comparison of Techniques. Cognit Comput. 2016;8:757-771. doi: 10.1007/s12559-016-9415-7. Epub 2016 Jun 1. PMID: 27563360; PMCID: PMC4981629.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Duwairi, R. M., & Qarqaz, I. (2014, August). Arabic sentiment analysis using supervised classification. In 2014 International Conference on Future Internet of Things and Cloud (pp. 579-583). IEEE.

ElJundi, O., Antoun, W., El Droubi, N., Hajj, H., El-Hajj, W., & Shaban, K. (2019, August). hulmona: The universal language model in arabic. In Proceedings of the fourth Arabic natural language processing workshop (pp. 68-77).

Elshakankery, K., & Ahmed, M. F. (2019). HILATSA: A hybrid Incremental learning approach for Arabic tweets sentiment analysis. *Egyptian Informatics Journal*, *20*(3), 163-171.

Farha, I. A., & Magdy, W. (2019, August). Mazajak: An online Arabic sentiment analyser. In Proceedings of the Fourth Arabic Natural Language Processing Workshop (pp. 192-198).

Gautam, G., & Yadav, D. (2014, August). Sentiment analysis of twitter data using machine learning approaches and semantic analysis**.** In 2014 Seventh International Conference on Contemporary Computing (IC3) (pp. 437-442). IEEE.

Heikal, M., Torki, M., & El-Makky, N. (2018). Sentiment analysis of Arabic tweets using deep learning. Procedia Computer Science, 142, 114-122.

Kanan, T., Sadaqa, O., Aldajeh, A., Alshwabka, H., AlZu'bi, S., Elbes, M., ... & Alia, M. A. (2019, April). A review of natural language processing and machine learning tools used to analyze arabic social media. In 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) (pp. 622-628). IEEE.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lin, Y., Meng, Y., Sun, X., Han, Q., Kuang, K., Li, J., & Wu, F. (2021). Bertgcn: Transductive text classification by combining gcn and bert. arXiv preprint arXiv:2105.05727.

Lu, Z., Du, P., & Nie, J. Y. (2020, April). VGCN-BERT: augmenting BERT with graph embedding for text classification. In *European Conference on Information Retrieval* (pp. 369-382). Springer, Cham.

Mahdaouy, A. E., Mekki, A. E., Essefar, K., Mamoun, N. E., Berrada, I., & Khoumsi, A. (2021). Deep multi-task model for sarcasm detection and sentiment analysis in arabic language. *arXiv preprint arXiv:2106.12488*.

Neethu, M. S., & Rajasree, R. (2013). Sentiment analysis in twitter using machine learning techniques. In 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

Obeid, O., Zalmout, N., Khalifa, S., Taji, D., Oudah, M., Alhafni, B., ... & Habash, N. (2020, May). CAMeL tools: An open source python toolkit for Arabic natural language processing. In Proceedings of the 12th language resources and evaluation conference (pp. 7022-7032).

Ombabi, A. H., Ouarda, W., & Alimi, A. M. (2020). Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks. Social Network Analysis and Mining, 10(1), 1-13.

Sayed, A. A., Elgeldawi, E., Zaki, A. M., & Galal, A. R. (2020, February).

Shoukry, A., & Rafea, A. (2012, May). Sentence-level Arabic sentiment analysis. In 2012 International Conference on Collaboration Technologies and Systems (CTS) (pp. 546-550). IEEE.

Song, B., Pan, C., Wang, S., & Luo, Z. (2021, April). DeepBlueAI at WANLP-EACL2021 task 2: A Deep Ensemble-based Method for Sarcasm and Sentiment Detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop* (pp. 390-394).

Wadhawan, A. (2021). Arabert and farasa segmentation based approach for sarcasm and sentiment detection in arabic tweets. *arXiv preprint arXiv:2103.01679*.

Yao, L., Mao, C., & Luo, Y. (2019, July). Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 7370-7377).

# الملخص

تحليل المشاعر للنص العربي مجال ذو اهتمام كبير في السنوات الأخيرة. تم استخدام العديد من الطرق في تحليل المشاعر العربية. ومع ذلك، فإن قدرة النماذج المقترحة التقاط المعلومات العالمية حول البيانات على سياق المعلومات المضمنة في الجملة. نبين في هذه الرسالة قوة استخدام شبكة المخططات للرسم البياني (GCN) في التقاط المعنى العالمي للكلمة وتدمج الرسم البياني مع نموذج MARBERT أحد أكثر النماذج تقدمًا في معالجة اللغة العربية لتحقيق دقة أعلى لمجموعات بيانات تويتر العربية المختلفة. يستخدم نموذج GCN-MARBERT طريقة MARBERT لترميز الكلمات ويقوم ببناء شبكة عصبية للرسم البياني تعتمد على معلومات متبادلة لبناء علاقة بين مفردات مجموعة البيانات كاملة وسيتم تمرير هذا الرسم البياني من خلال آلية الانتباه (attention) إلى مصنف النموذج الخاص بي حتى يقوم بعمل تصنيفات للتغريدات مع الحفاظ على المعلومات العالمية حول المفردات. تم تدريب النموذج المقترح على ثلاث مجموعات بيانات وهي: ASTD, AJGT و ArSarcasm . بلغت الدقة التي تم تحقيقها لمجموعات البيانات هذه: 74.2٪ و 97.2٪ و 72.4٪ على التوالي. وهذا يظهر كفاءة استخدام الشبكات العصبية للرسم البياني في تحليل المشاعر وتصنيف النص وقدرتها كنموذج أساسي في معالجة اللغة العربية.